BIOQUERY-ASP: Querying Biomedical Databases and Ontologies using Answer Set Programming

Esra Erdem and Umut Oztok

Sabancı University, İstanbul, Turkey
Motivation

- Biomedical data is stored in various structured forms and at different locations.
- With the current Web technologies, reasoning over these data is limited to answering simple queries by keyword search and by some direction of humans.
- Vital research, like drug discovery, requires deep reasoning (e.g., answering complex queries, generating explanations).
Complex Queries

Q1 What are the genes that are targeted by the drug Epinephrine and that interact with the gene DLG4?

Q2 What are the genes that are targeted by all the drugs that belong to the category Hmg-coa reductase inhibitors?

Q3 What are the cliques of 5 genes, that contain the gene DLG4?

Q4 What are the genes that are related to the gene ADRB1 via a gene-gene relation chain of length at most 3?

Q5 What are the most similar 3 genes that are targeted by the drug Epinephrine?
Challenges

- It is hard to represent a query in a formal language.

- Complex queries require recursive definitions, aggregates, etc..

- Databases/ontologies are in different formats/locations.

- Databases/ontologies are large.

- Experts may ask for further explanations.
Challenges

- It is hard to represent a query in a formal language.
  - Represent queries in a controlled natural language (CNL) – BIOQUERY-CNL* [EY09, EEO11].
- Complex queries require recursive definitions, aggregates, etc..
  - Represent queries in Answer Set Programming (ASP) [BCD⁺08, EEO11].
- Databases/ontologies are in different formats/locations.
  - Integration of knowledge via a rule layer in ASP [BCD⁺08, EEO11].
- Databases/ontologies are large.
  - Extract the relevant part for faster reasoning [EEO11].
- Experts may ask for further explanations.
  - Algorithm for generating shortest/different explanations [EEO11].
BIOQUERY-ASP: System Overview

User Interface
- Query in Natural Language
- Query in ASP

Databases/Ontologies
- Rule Layer in ASP

Query Answering
- Relevant Knowledge in ASP
- Query in ASP

Answer

Explanation Generation
- Explanation in ASP
- Explanation in Natural Language

Related Webpages
Answer Set Programming (ASP)

- Knowledge representation and automated reasoning paradigm.
- Theoretical basis: answer set semantics (Gelfond & Lifschitz, 1988).
- Expressive representation language: Defaults, recursive definitions, aggregates, preferences, etc.
- ASP solvers:
  - SMODELS (Helsinki University of Technology, 1996)
  - DLV (Vienna University of Technology, 1997)
  - CMODELS (University of Texas at Austin, 2002)
  - PBMODELS (University of Kentucky, 2005)
  - CLASP (University of Potsdam, 2006) – winning first places at ASP’07/09/11/12, PB’09/11/12, and SAT’09/11/12
Applications of ASP in Artificial Intelligence

- planning ([Lif02], [DEF^03], [SPS09], [TSGM11], [GKS12])
- theory update/revision ([IS95], [FGP07], [OC07], [EW08], [ZCRO10], [Del10])
- preferences ([SW01], [Bre07], [BNT08])
- diagnosis ([EFLP99], [BG03], [EBDT^09])
- learning ([Sak01], [Sak05], [SI09], [CSIR11])
- description logics and semantic web ([EGRH06], [CEO09], [Sim09], [PHE10], [SW11], [EKSX12])
- probabilistic reasoning ([BH07], [BGR09])
- data integration and question answering ([AFL10], [LGI^05])
- multi-agent systems ([VCP^05], [SPS09], [SS09], [BGSP10], [Sak11], [PSBG12])
- multi-context systems ([EBDT^09], [BEF11], [EFS11], [BEFW11], [DFS12])
- natural language processing/understanding ([BDS08], [BGG12], [LS12])
- argumentation ([EGW08], [WCG09], [EGW10], [Gag10])
Applications of ASP in Other Areas

- product configuration ([SN98], [TSNS03])
- Linux package configuration ([Syr00], [GKS11])
- wire routing ([ELW00], [ET01])
- combinatorial auctions ([BU01])
- game theory ([VV02], [VV04])
- decision support systems ([NBG+01])
- logic puzzles ([FMT02], [BD12])
- bioinformatics ([BCD+08], [EY09], [EEB10], [EEE011])
- phylogenetics ([ELR06], [BEE+07], [Erd09], [EEEF09], [CEE11], [Erd11])
- haplotype inference ([EET09], [TE08])
- systems biology ([TB04], [GGI+10], [ST09], [TAL+10], [GSTV11])
- automatic music composition ([BBVF09], [BBVF11])
- assisted living ([MMB08], [MMB09], [MSMB11])
- team building ([RGA+12])
- robotics ([CHO+09], [EHP+11], [AEEP11], [EHPU12], [APE12])
- software engineering ([EIO+11])
- bounded model checking ([HN03], [TT07])
- verification of cryptographic protocols ([DGH09])
- e-tourism ([RDG+10])
Applications of ASP in Other Areas

- product configuration ([SN98], [TSNS03]): used by Variantum Oy
- Linux package configuration ([Syr00], [GKS11])
- wire routing ([ELW00], [ET01])
- combinatorial auctions ([BU01])
- game theory ([VV02], [VV04])
- decision support systems ([NBG+01]): used by United Space Alliance
- logic puzzles ([FMT02], [BD12])
- bioinformatics ([BCD+08], [EY09], [EEB10], [EEE011])
- phylogenetics ([ELR06], [BEE+07], [Erd09], [EEEF09], [CEE11], [Erd11])
- haplotype inference ([EET09], [TE08])
- systems biology ([TB04], [GGI+10], [ST09], [TAL+10], [GSTV11])
- automatic music composition ([BBVF09], [BBVF11])
- assisted living ([MMB08], [MMB09], [MSMB11])
- team building ([RGA+12]): used by Gioia Tauro seaport
- robotics ([CHO+09], [EHP+11], [AEEP11], [EHPU12], [APE12])
- software engineering ([EIO+11])
- bounded model checking ([HN03], [TT07])
- verification of cryptographic protocols ([DGH09])
- e-tourism ([RDG+10])
**BIOQUERY-ASP: System Overview**

- **User Interface**
  - Query in Natural Language
  - Query in ASP

- **Databases/Ontologies**
  - Rule Layer in ASP

- **Query Answering**
  - Relevant Knowledge in ASP
  - Query in ASP

- **Explanation Generation**
  - Explanation in ASP
  - Explanation in Natural Language

- **Answer**
  - Related Webpages
**BioQuery-CNL*: A CNL for biomedical queries**

**BioQuery-CNL* Grammar:**

QUERY → WHATQUERY QUESTIONMARK
WHATQUERY → What are OFRELATION NESTEDPREDICATERELATION
OFRELATION → Noun() of Type()
NESTEDPREDICATERELATION → (...)∗ that PREDICATERELATION
PREDICATERELATION → INSTANCERELATION (...)∗
INSTANCERELATION → (NEG)? Verb() the Type() Instance()
QUESTIONMARK → ?

**Ontology functions:**

Type() returns the type information, e.g., gene, disease, drug
Instance(T) returns instances of the type T, e.g., Asthma for type disease
Verb(T, T’) returns the verbs where type T is the subject and type T’ is the object, e.g., drug treat disease
Noun(T) returns the nouns that are related to the type T, e.g., side-effects of type drug

**Example:** What are the side-effects of the drugs that treat the disease Asthma?
Query Q2 in BIOQUERY-CNL*: What are the genes that are targeted by all the drugs that belong to the category Hmg-coa reductase inhibitors?

Query Q2 in ASP:

\[
\text{notcommon}(g_{n_1}) \leftarrow \text{not drug_gene}(d_2, g_{n_1}), \text{condition}_1(d_2) \\
\text{condition}_1(d) \leftarrow \text{drug_category}(d, \text{"Hmg – coa reductase inhibitors"}) \\
\text{what_be_gens}(g_{n_1}) \leftarrow \text{not notcommon}(g_{n_1}), \text{notcommon_exists} \\
\text{notcommon_exists} \leftarrow \text{notcommon}(x) \\
\text{answer_exists} \leftarrow \text{what_be_gens}(g_{n})
\]
Knowledge from RDF(S)/OWL ontologies can be extracted using “external predicates” supported by the ASP solver DLVHEX [EGRH06]:

\[
\text{triple\_gene}(x, y, z) \leftarrow \text{&\text{rdf["URIforGeneOntology"]}(x, y, z)}
\]

\[
\text{gene\_gene}(g_1, g_2) \leftarrow \text{triple\_gene}(x, \text{"geneproperties : name"}, g_1),
\text{triple\_gene}(x, \text{"geneproperties : related\_genes"}, b), \ldots
\]

ASP rules integrate the extracted knowledge, or define new concepts:

\[
\text{gene\_reachable\_from}(x, 1) \leftarrow \text{gene\_gene}(x, y), \text{start\_gene}(y)
\]
\[
\text{gene\_reachable\_from}(x, n + 1) \leftarrow \text{gene\_gene}(x, z),
\text{gene\_reachable\_from}(z, n), \text{max\_chain\_length}(l) \quad (0 < n, n < l)
\]
Generally, only a small part of the underlying databases/ontologies and the rule layer is related to the given query.

We introduce a method to identify the relevant part of the ASP program for more efficient query answering.
% Databases and Ontologies:
fact 1.
fact 2.
fact 3.
...

% Rule Layer:
rule 1.
rule 2.
rule 3.
...
Identifying the Relevant Part of a Program

% Databases and Ontologies:
  fact 1.
  fact 2.
  fact 3.
  ...

% Rule Layer:
  rule 1.
  rule 2.
  rule 3.
  ...

% Query:
  rule 1.
  rule 2.
  ...
  ...
Identifying the Relevant Part of a Program

% Databases and Ontologies:
  fact 1.
  fact 2.
  fact 3.
  ...

% Rule Layer:
  rule 1.
  rule 2.
  rule 3.
  ...

% Query:
  rule 1.
  rule 2.
  ...

Esra Erdem and Umut Oztok
## Experimental Results: Databases & Ontologies

<table>
<thead>
<tr>
<th>Source</th>
<th>Relation (number of ASP facts)</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>BIOGRID</strong></td>
<td>gene-gene (372.293)</td>
</tr>
<tr>
<td><strong>DRUGBANK</strong></td>
<td>drug-drug (21.756)</td>
</tr>
<tr>
<td></td>
<td>drug-category (4.743)</td>
</tr>
<tr>
<td><strong>SIDER</strong></td>
<td>drug-sideeffect (61.102)</td>
</tr>
<tr>
<td><strong>PHARMGKB</strong></td>
<td>drug-disease (3.740)</td>
</tr>
<tr>
<td></td>
<td>drug-gene (15.805)</td>
</tr>
<tr>
<td></td>
<td>disease-gene (9.417)</td>
</tr>
<tr>
<td><strong>CTD</strong></td>
<td>drug-disease (704.590)</td>
</tr>
<tr>
<td></td>
<td>drug-gene (259.048)</td>
</tr>
<tr>
<td></td>
<td>disease-gene (8.909.071)</td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td>10.3 M</td>
</tr>
</tbody>
</table>
## Experimental Results

<table>
<thead>
<tr>
<th>Query</th>
<th>Complete</th>
<th>Relevant</th>
</tr>
</thead>
<tbody>
<tr>
<td>Q1</td>
<td>271.39</td>
<td>13.08</td>
</tr>
<tr>
<td></td>
<td>Rules: 21059323</td>
<td>Rules: 1961789</td>
</tr>
<tr>
<td>Q2</td>
<td>266.06</td>
<td>14.34</td>
</tr>
<tr>
<td></td>
<td>Rules: 21059909</td>
<td>Rules: 2084579</td>
</tr>
<tr>
<td>Q3</td>
<td>266.62</td>
<td>9.85</td>
</tr>
<tr>
<td></td>
<td>Rules: 21059248</td>
<td>Rules: 1567401</td>
</tr>
<tr>
<td>Q4</td>
<td>273.93</td>
<td>321.11</td>
</tr>
<tr>
<td></td>
<td>Rules: 21059353</td>
<td>Rules: 19450525</td>
</tr>
<tr>
<td>Q5</td>
<td>265.91</td>
<td>9.93</td>
</tr>
<tr>
<td></td>
<td>Rules: 21061727</td>
<td>Rules: 1460831</td>
</tr>
<tr>
<td>Q6</td>
<td>269.69</td>
<td>320.56</td>
</tr>
<tr>
<td></td>
<td>Rules: 21111842</td>
<td>Rules: 19512500</td>
</tr>
<tr>
<td>Q7</td>
<td>270.05</td>
<td>6.07</td>
</tr>
<tr>
<td></td>
<td>Rules: 21062006</td>
<td>Rules: 1023061</td>
</tr>
<tr>
<td>Q8</td>
<td>275.19</td>
<td>7.02</td>
</tr>
<tr>
<td></td>
<td>Rules: 21079275</td>
<td>Rules: 1040406</td>
</tr>
<tr>
<td>Q9</td>
<td>272.48</td>
<td>3.48</td>
</tr>
<tr>
<td></td>
<td>Rules: 21059597</td>
<td>Rules: 547545</td>
</tr>
<tr>
<td>Q10</td>
<td>266.37</td>
<td>11.25</td>
</tr>
<tr>
<td></td>
<td>Rules: 21077252</td>
<td>Rules: 1594891</td>
</tr>
</tbody>
</table>
Example: Explanation Generation

**Query in** BIOQUERY-CNL*: What are the genes that are targeted by the drug Epinephrine and that interact with the gene DLG4?

**An Answer:** ADRB1

**Shortest Explanation in ASP:**

```
what_be_genes(ADRB1) ← drug_gene(Epinephrine, ADRB1), gene_gene(ADRB1, DLG4)
```

**Explanation in Natural Language:**

The drug Epinephrine targets the gene ADRB1 according to CTD.
The gene DLG4 interacts with the gene ADRB1 according to BioGrid.
BioQuery-ASP

User Interface
Query in Natural Language
Query in ASP

Databases/Ontologies
Rule Layer in ASP

Query Answering
Relevant Knowledge in ASP
Query in ASP

Answer

Explanation Generation
Explanation in ASP
Explanation in Natural Language

Related Webpages

http://krr.sabanciuniv.edu/projects/BioQuery-ASP/
Related Publications


- E. Erdem and R. Yeniterzi: *Transforming Controlled Natural Language Biomedical Queries into Answer Set Programs*, *Proc. of BioNLP’09*.


- U. Oztok and E. Erdem: *Generating Explanations for Complex Biomedical Queries*, *Proc. of AAAI’11*.

- E. Erdem, H. Erdogan, and U. Oztok: *BIOQUERY-ASP: Querying Biomedical Ontologies using Answer Set Programming*, *Proc. of RuleML’11@BRF Challenge*. 
Erdi Aker, Ahmetcan Erdogan, Esra Erdem, and Volkan Patoglu.
Causal reasoning for planning and coordination of multiple housekeeping robots.

M. Alviano, W. Faber, and N. Leone.
Disjunctive asp with functions: Decidable queries and effective computation.

Erdi Aker, Volkan Patoglu, and Esra Erdem.
Answer set programming for reasoning with semantic knowledge in collaborative housekeeping robotics.
In *Proc. of the 10’th IFAC Symposium on Robot Control (SYROCO)*, 2012.

G. Boenn, M. Brain, M. D. Vos, and J. Fitch.
Anton: Composing logic and logic composing.

Georg Boenn, Martin Brain, Marina De Vos, and John Fitch.
Automatic music composition using answer set programming.

Olivier Bodenreider, Zeynep H. Coban, Mahir C. Doganay, Esra Erdem, and Hilal Kosucu.
A preliminary report on answering complex queries related to drug discovery using answer set programming.
Chitta Baral and Juraj Dzifcak.
Solving puzzles described in english by automated translation to answer set programming and learning how to do that translation.

Chitta Baral, Juraj Dzifcak, and Tran Cao Son.
Using answer set programming and lambda calculus to characterize natural language sentences with normatives and exceptions.

Inferring phylogenetic trees using answer set programming.

Gerhard Brewka, Thomas Eiter, and Michael Fink.
Nonmonotonic multi-context systems: A flexible approach for integrating heterogeneous knowledge sources.

Gerhard Brewka, Thomas Eiter, Michael Fink, and Antonius Weinzierl.
Managed multi-context systems.

M. Balduccini and M. Gelfond.
Diagnostic reasoning with a-prolog.
Chitta Baral, Marcos Alvarez Gonzalez, and Aaron Gottesman.

The inverse lambda calculus algorithm for typed first order logic lambda calculus and its application to translating english to fol.


Probabilistic reasoning with answer sets.

Chitta Baral, Gregory Gelfond, Tran Cao Son, and Enrico Pontelli.

Using answer set programming to model multi-agent scenarios involving agents' knowledge about other's knowledge.


Chitta Baral and Matt Hunsaker.

Using the probabilistic logic programming language p-log for causal and counterfactual reasoning and non-naive conditioning.


Gerhard Brewka, Ilkka Niemelä, and Miroslaw Truszczynski.

Preferences and nonmonotonic reasoning.

G. Brewka.

Preferences, contexts and answer sets.
C. Baral and C. Uyan.
Declarative specification and solution of combinatorial auctions using logic programming.

D. Cakmak, E. Erdem, and H. Erdogan.

D. Calvanese, T. Eiter, and M. Ortiz.
Regular path queries in expressive description logics with nominals.

Ozan Caldiran, Kadir Haspalamutgil, Abdullah Ok, Can Palaz, Esra Erdem, and Volkan Patoglu.
Bridging the gap between high-level reasoning and low-level control.

Domenico Corapi, Daniel Sykes, Katsumi Inoue, and Alessandra Russo.
Probabilistic rule learning in nonmonotonic domains.

Monitoring agents using declarative planning.

James P. Delgrande.
A program-level approach to revising logic programs under the answer set semantics.
Jürgen Dix, Wolfgang Faber, and V. S. Subrahmanian.
Privacy preservation using multi-context systems and default logic.

J. P. Delgrande, T. Grote, and A. Hunter.
A general approach to the verification of cryptographic protocols using answer set programming.

Combining nonmonotonic knowledge bases with external sources.

Halit Erdogan, Esra Erdem, and Olivier Bodenreider.
Exploiting umls semantics for checking semantic consistency among umls concepts.

T. Eiter, E. Erdem, H. Erdogan, and M. Fink.
Finding similar or diverse solutions in answer set programming.

Esra Erdem, Yelda Erdem, Halit Erdogan, and Umut Oztok.
Finding answers and generating explanations for complex biomedical queries.

Esra Erdem, Halit Erdogan, and Umut Oztok.
Bioquery-as: Querying biomedical ontologies using answer set programming.
In Proc. of RuleML2011@BRF Challenge, 2011.
E. Erdem, O. Erdem, and F. Türe.
Haplo-asp: Haplotype inference using answer set programming.

T. Eiter, W. Faber, N. Leone, and G. Pfeifer.
The diagnosis frontend of the dlv system.

Thomas Eiter, Michael Fink, and Peter Schüller.
Approximations for explanations of inconsistency in partially known multi-context systems.

Thomas Eiter, G.Ianni, R.Schindlauer, and H.Tompits.
Effective integration of declarative rules with external evaluations for Semantic-Web reasoning.

Uwe Egly, Sarah Alice Gaggl, and Stefan Woltran.
Aspartix: Implementing argumentation frameworks using answer-set programming.

Uwe Egly, Sarah Alice Gaggl, and Stefan Woltran.
Answer-set programming encodings for argumentation frameworks.

Combining high-level causal reasoning with low-level geometric reasoning and motion planning for robotic manipulation.
Esra Erdem, Kadir Haspalamutgil, Volkan Patoglu, and Tansel Uras.
Causality-based planning and diagnostic reasoning for cognitive factories.

Esra Erdem, Katsumi Inoue, Johannes Oetsch, Joerg Puehrer, Hans Tompits, and Cemal Yilmaz.
Answer-set programming as a new approach to event-sequence testing.

Thomas Eiter, Thomas Krennwallner, Patrik Schneider, and Guohui Xiao.
Uniform evaluation of nonmonotonic dl-programs.

E. Erdem, V. Lifschitz, and D. Ringe.
Temporal phylogenetic networks and logic programming.

E. Erdem, V. Lifschitz, and M. F. Wong.
Wire routing and satisfiability planning.

E. Erdem.
Phylo-asp: Phylogenetic systematics with answer set programming.

Esra Erdem.
Applications of answer set programming in phylogenetic systematics.
D. East and M. Truszczynski.
More on wire routing with asp.

Thomas Eiter and Kewen Wang.
Semantic forgetting in answer set programming.

Esra Erdem and Reyyan Yeniterzi.
Transforming controlled natural language biomedical queries into answer set programs.

Fernando Zacarias Flores, Mauricio Javier Osorio Galindo, and Edgar Fernandez Plascencia.
Updates under pstable.

R. Finkel, V. W. Marek, and M. Truszczynski.
Constraint lingo: A program for solving logic puzzles and other tabular constraint problems, 2002.

Sarah Alice Gaggl.
Towards a general argumentation system based on answer-set programming.
In ICLP (Technical Communications), pages 265–269, 2010.

Repair and prediction (under inconsistency) in large biological networks with answer set programming.
M. Gebser, R. Kaminski, and T. Schaub.

aspccd: A Linux package configuration tool based on answer set programming.

Martin Gebser, Roland Kaufmann, and Torsten Schaub.

Gearing up for effective asp planning.


Detecting inconsistencies in large biological networks with answer set programming.

K. Heljanko and I. Niemela.

Bounded LTL model checking with stable models.

K. Inoue and C. Sakama.

Abductive framework for nonmonotonic theory change.
In IJCAI, pages 204–210, 1995.


The infomix system for advanced integration of incomplete and inconsistent data.

V. Lifschitz.

Answer set programming and plan generation.
Yuliya Lierler and Peter Schüßler.
Parsing combinatory categorial grammar via planning in answer set programming.

A. Mileo, D. Merico, and R. Bisiani.
Wireless sensor networks supporting context-aware reasoning in assisted living.

A. Mileo, D. Merico, and R. Bisiani.
Non-monotonic reasoning supporting wireless sensor networks for intelligent monitoring: The sindi system.

Knowledge-based multi-criteria optimization to support indoor positioning.

M. Nogueira, M. Balduccini, M. Gelfond, R. Watson, and M. Barry.
An a-prolog decision support system for the space shuttle.

Mauricio Osorio and Victor Cuevas.
Updates in answer set programming: An approach based on basic structural properties.

Jörg Pührer, Stijn Heymans, and Thomas Eiter.
Dealing with inconsistency when combining ontologies and rules using dl-programs.
Enrico Pontelli, Tran Cao Son, Chitta Baral, and Gregory Gelfond.
Answer set programming and planning with knowledge and world-altering actions in multiple agent domains.

Francesco Ricca, Antonella Dimasi, Giovanni Grasso, Salvatore Maria Ielpa, Salvatore Iiritano, Marco Manna, and Nicola Leone.
A logic-based system for e-tourism.

Francesco Ricca, Giovanni Grasso, Mario Alviano, Marco Manna, Vincenzino Lio, Salvatore Iiritano, and Nicola Leone.
Team-building with answer set programming in the gioia-tauro seaport.

C. Sakama.
Learning by answer sets.

Chiaki Sakama.
Induction from answer sets in nonmonotonic logic programs.

Chiaki Sakama.
Dishonest reasoning by abduction.

Chiaki Sakama and Katsumi Inoue.
Brave induction: a logical framework for learning from incomplete information.
Mantas Simkus.
Fusion of logic programming and description logics.

T. Soininen and I. Niemelä.
Developing a declarative rule language for applications in product configuration.

T. C. Son, E. Pontelli, and C. Sakama.
Logic programming for multiagent planning with negotiation.

T. C. Son and C. Sakama.
Reasoning and planning with cooperative actions for multiagents using answer set programming.

T. Schaub and S. Thiele.
Metabolic network expansion with answer set programming.

A comparative study of logic programs with preference.

Yi-Dong Shen and Kewen Wang.
Extending logic programs with description logic expressions for the semantic web.
Tommi Syrjänen.
Including diagnostic information in configuration models.

Luis Tari, Saadat Anwar, Shanshan Liang, Jörg Hakenberg, and Chitta Baral.
Synthesis of pharmacokinetic pathways through knowledge acquisition and automated reasoning.

N. Tran and C. Baral.
Reasoning about triggered actions in ansprolog and its application to molecular interactions in cells.

F. Türe and E. Erdem.
Efficient haplotype inference with answer set programming.

Phan Huy Tu, Tran Cao Son, Michael Gelfond, and A. Ricardo Morales.
Approximation of action theories and its application to conformant planning.

Juha Tiihonen, Timo Soininen, Ilkka Niemelae, and Reijo Sulonen.
A practical tool for mass-customising configurable products.

Calvin Kai Fan Tang and Eugenia Ternovska.
Model checking abstract state machines with answer set programming.
A multi-agent platform using ordered choice logic programming.
In *Declarative Agent Languages and Technologies (DALT’05)*, pages 72–88, 2005.

Marina De Vos and Dirk Vermeir.
Dynamic decision-making in logic programming and game theory.

M. D. Vos and D. Vermeir.
Extending answer sets for logic programming agents.

Yining Wu, Martin Caminada, and Dov M. Gabbay.
Complete extensions in argumentation coincide with 3-valued stable models in logic programming.

Claudia Zepeda, José Luis Carballido, Mario Rossainz, and Mauricio Osorio.
Updates based on asp.
In *Proc. of MICAI (Special Sessions)*, pages 63–66, 2010.