# Answer Set Programs with Queries over Subprograms

Motivation Answer Set Programming is a well-known declarative problem solving approach.

### Christoph Redl

redl@kr.tuwien.ac.at





1 Motivation

2 The Sat

- 3 Deciding
- 4 Query An
- 5 Discussion
- 6 Conclusion

# Answer Set Programming is a well-known declarative problem solving approach.

Motivation

Answer Set Programming [Gelfond and Lifschitz, 1991]

 $a_1 \lor \cdots \lor a_n \leftarrow b_1, \ldots, b_m, \text{not } b_{m+1}, \ldots, \text{not } b_n,$ An ASP program consists of rules of form

it is an answer set of a ground program P, if I is a  $\subseteq$ -minimal model of the reduct  $P^I=\{H(r)\leftarrow B^+(r)\mid r\in\Pi,I\not\models b \text{ for all }b\in B^-(r)\}.$ An interpretation I is a set of ground atoms;

Semantics of non-ground programs is defined via a grounding, i.e., replacement of all variables by all constants in all possible ways.

### Motivation

Answer Set Programming is a well-known declarative problem solving approach.

 $a_1 \vee \cdots \vee a_n \leftarrow b_1, \ldots, b_m, \text{not } b_{m+1}, \ldots, \text{not } b_n,$ An ASP program consists of rules of form

Answer Set Programming [Gelfond and Lifschitz, 1991]

An interpretation I is a set of ground atoms; it is an answer set of a ground program  $P_i$  if I is a  $\subseteq$ -minimal model of the reduct  $P^i = \{H(r) \leftarrow B^+(r) \mid r \in \Pi, I \not\equiv b$  for all  $b \in B^-(r)\}$ .

Semantics of non-ground programs is defined via a grounding, i.e., replacement of all variables by all constants in all possible ways.

### Two (Related) Restrictions

■ Meta-reasoning about the answer sets of a (sub)program within another (meta-)program not inherently supported.

Motivation Answer Set Programming is a well-known declarative problem solving approach.

An ASP program consists of rules of form  $a_1 \vee \dots \vee a_n \leftarrow b_1, \dots, b_m, \text{not } b_{m+1}, \dots, \text{not } b_n,$ 

Answer Set Programming [Gelfond and Lifschitz, 1991]

An interpretation I is a set of ground atoms; it is an answer set of a ground program  $P_i$  if I is a  $\subseteq$ -minimal model of the reduct  $P^I=\{H(r)\leftarrow B^+(r)\mid r\in\Pi, I\not\models b$  for all  $b\in B^-(r)\}.$ 

Semantics of non-ground programs is defined via a grounding, i.e., replacement of all variables by all constants in all possible ways.

### Two (Related) Restrictions

- Meta-reasoning about the answer sets of a (sub)program within another (meta-)program not inherently supported.
- Despite  $\Sigma_r^2$ -completeness of disjunctive ASP, solving problems from the first level of the polynomial hierarchie is sometimes tricky.

Motivation  Two (Related) Restrictions  ■ Meta-reasoning about the answer sets of a ( <i>sub</i> ) <i>program</i> within another (meta-)program not inherently supported. ■ Despite ∑²² completeness of disjunctive ASP; solving problems from the first level of the polynomial hierarchie within a program is difficult.	d) Restrictions  soning about the answer sets of a (sub)program within another  gram not inherently supported.  f-completeness of disjunctive ASR; solving problems from the first e polynomial herarchie within a program is difficult.  ing to decide inconsistency of a normal program within a e) program.	Motivation  Two (Related) Restrictions  Weta-reasoning about the answer sets of a (sub)program within another (meta-)program not inherently supported.  Despite 5½-completeness of disjunctive ASP, solving problems from the first level of the polynomial hierarchie within a program is difficult.  Contribution  An encoding to decide inconsistency of a normal program within a (disjunctive) program.  An encoding for query answering over a normal program within another program.
Inswer sets of a (sub)program within another thy supported.  thy supported.  the disjunctive ASP, solving problems from the first archie within a program is difficult.	I) Restrictions oning about the answer sets of a (sub)program within another point gram not inherently supported.  **Completeness of disjunctive ASP, solving problems from the first polynomal hierarchie within a program is difficult.  **Grammal program within a program within a program.  **Program.**	I) Restrictions oning about the answer sets of a (sub)program within an gram not inherently supported. 'completeness of disjunctive ASP, solving problems from a polynomial hierarchie within a program is difficult.  g to decide inconsistency of a normal program within any of or query answering over a normal program within ang for query answering over a normal program within and for query answering over a normal program within and grown and the program within and gord query answering over a normal program within and gord query answering over a normal program within and gord query answering over a normal program within and gord query answering over a normal program within and gord query answering over a normal program within and gord query answering over a normal program within and gord query answering over a normal program within and gord query answering over a normal program within and gord query answering over a normal program within and gord query answering over a normal program within and gord query answering over a normal program within and gord query answering over a normal program within and gord query answering over a normal program within and gord query answering over a normal program within and gord query answering over a normal program within and gord query answering over a normal program within and gord query
first	oning about the answer sets of a (sub)program within another gram not inherently supported'completeness of disjunctive ASP solving problems from the first is polynomial hierarchie within a program is difficult.  go decide inconsistency of a normal program within a sprogram.	oning about the answer sets of a (sub)program within an gram not inherently supported.  'completeness of disjunctive ASP, solving problems from is polynomial hierarchie within a program is difficult.  g to decide inconsistency of a normal program within a normal program.  ng for query answering over a normal program within ance
1	g to decide inconsistency of a normal program within a s) program.	ing to decide inconsistency of a normal program within a program.  The for query answering over a normal program within and the control of th
(disip)	ELLS TRACE, NACE AND THE PROPERTY OF THE PROPE	nade (todera) recenques anno annotate ann
Fact C (TU Mores) HEX-Programs Advis 2017 4126 Red		
Motivation Outline	The Salacation Technique and dis regimences	The Saturation Technique
		Basic idea
another 2	ation Technique and its Restrictions	■ Exploits disjunctions with head-cycles to solve coNP-hard problems within ASP.
Despite 2.3 Completeness of disjunctive Aor; sowing problems from the first level of the polynomial hierarchie within a program is difficult.		
ing to decide inconsistency of a normal program within a ne) program.  Ing for query answering over a normal program within another		
A language extension of ASP program with query atoms     A language extension of ASP program with query atoms to be used as a new modeling technique.		

The Sahzadon Tedmique and its Restrictors	The Saturation Tochnique and its Positidoxins	The Sahzabon Tedmique and its Restrictions
The Saturation Technique	The Saturation Technique	The Saturation Technique
Basic idea	Basic idea	Basic idea
<ul> <li>Exploits disjunctions with head-cycles to solve coNP-hard problems within ASP.</li> <li>(Based on the hardness proof of disjunctive ASP [Eiter and Gottlob, 1995].)</li> </ul>	<ul> <li>Exploits disjunctions with head-cycles to solve coNP-hard problems within ASP.</li> <li>(Based on the hardness proof of disjunctive ASP [Elter and Gottlob, 1995].)</li> <li>Typical use case:         Check if a certain property holds for all objects in a certain domain.     </li> </ul>	<ul> <li>Exploits disjunctions with head-cycles to solve coNP-hard problems within ASP.</li> <li>(Based on the hardness proof of disjunctive ASP [Eiter and Gottlob, 1995].</li> <li>Typical use case:         Check if a certain property holds for all objects in a certain domain.     </li> </ul>
		Example
		Check if a graph is not 3-colorable.
Red C (TU Vorna) PER Programm	Pool C (TU Vernal) HEXTRoyams Liby 4, 2017 6 / 26	Red C. (TU Versa) HEX Brogame July 4, 2017 6/26
Profession Strategies and Strategies	To describe the second described on the second describ	The State of Marian and Marian an
The Saturation Technique	The Saturation Technique	The Saturation Technique
Basic idea	Basic idea	Restrictions
<ul> <li>Exploits disjunctions with head-cycles to solve coNP-hard problems within ASP.</li> <li>(Based on the adminess proof of disjunctive ASP [Eiter and Gottlob, 1995].)</li> <li>Typical use case:         <ul> <li>Check if a certain property holds for all objects in a certain domain.</li> </ul> </li> </ul>	<ul> <li>Exploits disjunctions with head-cycles to solve coNP-hard problems within ASP. (Based on the hardness proof of disjunctive ASP [Elter and Gottlob, 1995].)</li> <li>Typical use case: Check if a certain property holds for all objects in a certain domain.</li> </ul>	Atthough any problem in coNP can be polynomially reduced to brave reasoning over disjunctive ASP, the reduction is not always obvious.
Example	Example	
Check if a graph is not 3-colorable. Consider $P_{nuntxol} = F \cup P_{punsv} \cup P_{anced} \cup P_{ax}$ where $P_{puns} \cup P_{anced} \cup P_{ax}$ where $P_{puns} \cup P_{anced} \cap P_{ax}$ where $P_{puns} \cup P_{anced} \cap P_{ax}$ where $P_{anced} = P_{anced} \cap P_{anced} \cap P_{anced}$ where $P_{anced} = P_{anced} \cap P_{anced}$ where $P_{anced} \cap P_{anced}$ where $P_{anced} \cap P_{anced}$ is $P_{anced} \cap P_{anced}$ where $P_{anced} \cap P_{anced}$ i	Oheok if a graph is not 3-colorable. Consider $P_{monto} = F \cup P_{guero} \cup P_{clock} \cup P_{gue}$ where $P \cup P_{guero} \cup P_{clock} \cup P_{guero} \cup P_{clock} \cup P_{guero} \cup P_{gue$	
No. 21 (11 MAN PART PARTY PART	is has the answer set $I_{sur} = A(P_{monto,ol})$ iff the graph $F$ is not 3-colorable. Otherwise its answer sets are proper subsets of $I_{sur}$ and represent 3-colorings.	The Grillians State of State o
Mode C (10 Methy)	Action (IV Verify) newstraying control of the contr	(D.Monta)

The Salvention Technique and its Restrictions	The Cannation Technique and be Ratifictions	The Salvantion Technique and its Bountet from
The Saturation Technique	The Saturation Technique	The Saturation Technique
Restrictions	Restrictions	Restrictions
<ul> <li>■ Although any problem in coNP can be polynomially reduced to brave reasoning over disjunctive ASP, the reduction is not always obvious.</li> <li>■ In particular, the saturation encoding cannot use default-negation.</li> <li>⇒ Checks which involve default-negations must be rewritten.</li> </ul>	■ Although any problem in coNP can be polynomially reduced to brave reasoning over disjunctive ASP, the reduction is not always obvious. ■ In particular, the saturation encoding cannot use default-negation.  ⇒ Checks which involve default-negations must be rewritten.	<ul> <li>■ Although any problem in coNP can be polynomially reduced to brave reasoning over disjunctive ASP, the reduction is not always obvious.</li> <li>■ In particular, the saturation encoding cannot use default-negation.</li> <li>⇒ Checks which involve default-negations must be rewritten.</li> </ul>
	Example Check if a graph has no vertex cover $S$ with size $ S  \le k$ for some integer $k$ .	Example Check if a graph has no vertex cover $S$ with size $ S  \le k$ for some integer $k$ . Consider $P_n$ consisting of facts $F$ over node and edge and the following parts: $P_{max} = \{n(X) \lor out(X), F_mode(X)\}$ and $n(X), f_mode(X), f_mode(X)\}$ . $P_{max} = \{n(X) \leftarrow node(X), not in(Y), not in(Y), f_mode(X),, n'(X_{k+1}), X_k \neq X_k,, X_k \neq X_{k+1}\}$ .
The Saturation Technique  Restrictions  ■ Although any problem in coNP can be polynomially reduced to brave reasoning over disjunctive ASP, the reduction is not always obvious.  ■ In particular, the saturation encoding cannot use default-negation.  ⇒ Checks which involve default-negations must be rewritten.  Example  Check if a graph has no vertex cover 5 with size  S  ≤ k for some integer k.  Consider Provider and the following parts:	The Saturation Technique  Example Decide if a ground normal ASP program P is inconsistent.	The Saturation Technique  Example Dedde if a ground normal ASP program $P$ is inconsistent.  Attempt: $P = \{num(a) \lor (aloc(b)   e \land (P)) \mid r \in P\}$ $ \cup \{num(a) \lor (aloc(b)   e \land (P)) \mid r \in P\} $ $ \cup \{num(a) \lor (aloc(b)   e \lor (P)) \mid r \in P\} $ $ \cup \{num(a) \lor (aloc(b) \mid e \lor (P)) \mid r \in P\} $ $ \cup \{num(a) \lor (aloc(b) \mid e \lor (P)) \mid e \lor (P)\} \} $ $ \cup \{num(a) \lor (aloc(b) \mid e \lor (P)) \mid e \lor (P)\} \} $ $ \cup \{num(a) \lor (aloc(b) \mid e \lor (P)) \mid e \lor (P)\} \} $ $ \cup \{num(a) \lor (aloc(b) \vdash and S\} \{aloc(b) \vdash and S\} \{aloc(b)\} \} \} $ $ \cup \{num(a) \lor (aloc(b) \vdash and S\} \{aloc(b) \vdash and S\} \{aloc(b)\} \} \} $ $ \cup \{num(a) \lor (aloc(b) \vdash and S\} \{aloc(b) \vdash and S\} \{aloc(b)\} \} \} $ $ \cup \{num(a) \lor (aloc(b) \vdash and S\} \{aloc(b) \vdash and S\} \{aloc(b)\} \} \} \} $ $ \cup \{num(a) \lor (aloc(b) \vdash and S\} \{aloc(b) \vdash and S\} \{aloc(b) \vdash and S\} \{aloc(b) \vdash and S\} \} \} \} \} $
$P_{m,a} = \{ux \leftarrow atgr(X,Y), not in(X), not in(Y), unt \leftarrow in(X), \dots, in(Xk+1), X_1 \neq X_2, \dots, X_k \neq X_{k+1}\}$ $P_{m,a} = \{n(X) \leftarrow mat(X), uni, aut(X) \leftarrow mat(X), uni$ $This encoding does not work as desired because model I_{m,a} = A(P_{m}) is unstable.$		

July 4, 2017 8/26

#### 3 Deciding Inconsistency of Normal Programs in Disjunctive ASP 5 Discussion 0 -4 9 E 8 8 9 E However, the comparison of the least model of the reduct to the original guess in rule uses default-negation. Eliminating negation is not straightforward. $$\label{eq:controlled} \begin{split} &\cup \left\{ \operatorname{controlled}(o) \mid e \in B^+(v) \right\} \mid r \in P, a \in H(r) \right\} \\ &\cup \left\{ \operatorname{not} A^- \in \operatorname{fact}(a) \mid \operatorname{controlled}(a) \mid a \in A(P) \right\} \\ &\cup \left\{ \operatorname{not} A^- \cap \operatorname{not}(a) \text{ and tradel(a)} \mid a \in A(P) \right\} \\ &\cup \left\{ \operatorname{not}(a) \leftarrow \operatorname{not} X_1 \mid \operatorname{dat}(a) \leftarrow \operatorname{not} X_2 \mid a \in A(P) \right\} \\ &\cup \left\{ \operatorname{not}(a) \leftarrow \operatorname{not} X_2 \mid \operatorname{dat}(a) \leftarrow \operatorname{not} X_2 \mid a \in A(P) \right\} \\ &\cup \left\{ \operatorname{otheriset}(r) \leftarrow \operatorname{not} X_2 \mid a \in A(P) \right\} \end{split}$$ Decide if a ground normal ASP program P is inconsistent. $\cup \; \{inReduct(r) \leftarrow \{false(b) \mid b \in B^-(r)\} \mid r \in P\}$ The Saturation Technique $P' = \{true(a) \lor false(a) | a \in A(P)\}$ Example Attempt: E 8 8 7 6 8 C However, the comparison of the least model of the reduct to the original guess in rule uses default-negation. $$\label{eq:controller} \begin{split} &\cup \{(aas) aote(a) + ind cale(r), \{(aas) doub(o) \mid b \in B^T(r)\} \mid r \in P, a \in H(r)\} \\ &\cup \{and S - faire(a), kantshool(o) \mid a \in A(P)\} \\ &\cup \{and S - rand(a), and tandshool(o) \mid a \in A(P)\} \\ &\cup \{nne(a) + and S, faire(a) + and S \mid a \in A(P)\} \\ &\cup \{inRedact(r) \leftarrow nndS\} \mid a \in A(P)\} \end{split}$$ Decide if a ground normal ASP program P is inconsistent $\cup \left\{ inReduct(r) \leftarrow \{false(b) \mid b \in B^-(r)\} \mid r \in P \right\}$ $P' = \{true(a) \vee false(a) | a \in A(P)\}$ The Saturation Technique Example Attempt:

#### $\cup \ \{iar(X,I) \leftarrow aom(X), iar(I), noAS; \ niter(X,I) \leftarrow aom(X), iar(I), noAS\}$ $\cup \ \{inReduct(R) \leftarrow rade(R), noAS; \ outReduct(R) \leftarrow rade(R), noAS\}$ A Meta-Program for Propositional Programs if P is inconsistent, M ∪ M<sup>p</sup><sub>gr</sub> has exactly one answer set which contains noAS; and (2) if P is consistent, M \(\cup M\)\_p^m has at least one answer set and none of the answes sets of M<sup>P</sup> contains noAS. ■ Use a meta-program M to simulate an ASP solver in disjunctive ASP. lacktriangle Encode the ASP program P to check for inconsistency as facts $M_{gr}^{P}$ ■ Evaluate $M \cup M_{gr}^P$ to identify inconsistency or the answer sets of P. (Alternative encodings exist, see e.g. [Eiter and Polleres, 2006]). A Meta-Program for Propositional Programs For any ground normal logic program P, we have that Proposition Basic idea ■ Use a meta-program M to simulate an ASP solver in disjunctive ASP. ■ Encode the ASP program P to check for inconsistency as facts $M_{gr}^P$ . $\blacksquare$ Evaluate $M \cup M_{gr}^P$ to identify inconsistency or the answer sets of P . (Alternative encodings exist, see e.g. [Eiter and Polleres, 2006]). A Meta-Program for Propositional Programs Basic idea

# A Meta-Program for Propositional Programs

 $\cup \; \{bodyP(r,b) \mid r \in P, h \in B^+(r)\} \cup \{bodyN(r,b) \mid r \in P, h \in B^-(r)\}$ 

Definition For a ground normal logic program P we let:  $M_{p_i}^{F} = \{m(\varsigma) \mid 0 \le c \mid k(F)\}\} \cup \{head(r,h) \mid r \in P, h \in H(\varsigma)\}$ 

A Meta-Program for Non-Ground Programs

To lift the idea to non-ground programs, we exploit function symbols: predicates in the input program become function symbols in the meta-program.

# A Meta-Program for Non-Ground Programs

To lift the idea to non-ground programs, we exploit function symbols: predicates in the input program become function symbols in the meta-program.

 $\begin{aligned} & \text{Definition} \\ & \text{For a (ground or non-ground) normal logic program $P$ we let:} \\ & M_w^* & \text{elect}() & \text{Sc} & \text{c}_{\{P(y)\}} \cup \{\text{blow}(r(\vec{\theta}_1, \mu) \in P_{r(x)} \mid x, \theta \mid r(\theta_1, \theta_2) \mid r(\theta_1) \mid x, \theta_2)\} \\ & \text{u} & \text{elect}(r(\vec{\theta}_1, \mu) \neq \{\text{blow}(R, \theta_1) \mid x \in B^{1}(r)\}) \mid r \in P_{r} \in B^{1}(r)\} \\ & \text{u} & \text{elect}(r(\vec{\theta}_1, \mu) \neq \{\text{blow}(R, \theta_1) \mid x \in B^{1}(r)\} \mid r \in P_{r} \in B^{1}(r)\} \end{aligned}$ 

## A Meta-Program for Non-Ground Programs

To lift the idea to non-ground programs, we exploit function symbols: predicates in the input program become function symbols in the meta-program.

### Definition

For a (ground or non-ground) normal logic program P we let:  $M_{ij}^{c} = \{are(i) \mid s < c < f(r)\} \mid v \in P, h \in H(r)\} \}$   $\cup \{are(i) \mid s < c < f(r)\} \mid v \in P, h \in H(r)\}$   $\cup \{are(i) \mid s < c < f(r)\} \mid v \in P, h \in H(r)\}$   $\cup \{are(i) \mid s < f(r)\} \mid v \in P, h \in H(r)\}$   $\cup \{are(i) \mid s < f(r)\} \mid v \in P, h \in H(r)\}$ 

#### Proposition

 if P is inconsistent, M ∪ M<sup>P</sup><sub>iig</sub> has exactly one answer set which contains noAS; For any normal logic program P, we have that

(2) if P is consistent, M \cup M<sup>p</sup><sub>re</sub> has at least one answer set and none of the answer sets of M<sup>p</sup> contains noAS.

# A Meta-Program for Non-Ground Programs A Meta-Program for Non-Ground Programs

Let  $P = \{f : d(a); \ r_1 \colon q(X) \leftarrow d(X), \ \text{not} \ p(X) \colon r_2 \colon p(X) \leftarrow d(X), \ \text{not} \ q(X) \}.$ Example

 $\mathsf{Let}\,P = \{f : d(a); \ r_1 : q(X) \leftarrow d(X), \mathsf{not}\,p(X); \ r_2 : p(X) \leftarrow d(X), \mathsf{not}\,q(X)\}.$ 

Example

We have: 
$$\begin{split} M_{i,i}^{c} &= (tend(i,d(a)) \leftarrow; hend(i,(X),q(X)) \leftarrow hend(i,d(X))) \leftarrow hend(i,d(X)) \rightarrow hend(i,d(X)) \leftarrow hend(i,d(X)) \rightarrow h$$

One description of Alexanders	Constitution of functions	A new formation of the second
Outline  Mortvation  The Saturation Technique and its Restrictions  Deciding Inconsistency of Normal Programs in Disjunctive ASP  Ouery Answering over Subprograms  Discussion  Conclusion	Query Answering over Subprograms We reduce brave and cautious queries over subprograms to inconsistency checking.	Query Answering over Subprograms  We reduce brave and cautious queries over subprograms to inconsistency checking.  Observation:  Proposition  For a normal logic program P and a query $q$ we have that  (1) $P \models_{p} q$ iff $P \cup \{ \leftarrow \overline{q} \mid I \in q \}$ is consistent; and  (2) $P \models_{c} q$ iff $P \cup \{ \leftarrow q \}$ is inconsistent.
Onstatement and discount	Only Assessiyona Gologotes	Our/Attents and Edingson
Query Answering over Subprograms We reduce brave and cautious queries over subprograms	Query Answering: Language Extension	Query Answering: Language Extension
to inconsistency checking. Observation:	Now we extend ASP with queries over subprograms:	Now we extend ASP with queries over subprograms:
Proposition	A ground query atom is of form $S \vdash_t q$ , where $t \in \{b,c\}$ determines the type of the	A ground query atom is of form $S \vdash_t q$ , where $t \in \{b,c\}$ determines the type of the
For a normal logic program $P$ and a query $q$ we have that (1) $P \models_b q$ iff $P \cup \{\leftarrow I \mid I \in q\}$ is consistent; and (2) $P \models_c q$ iff $P \cup \{\leftarrow q\}$ is inconsistent.	query, $\delta$ is a normal logic (sub)program, and $q$ is a query over $\delta$ .  Query atoms may occur in bodies of ASP programs in place of ordinary atoms.	query, 3 is a normal logic (sub)program, and $q$ is a query over 3.  Query atoms may occur in bodies of ASP programs in place of ordinary atoms.  Intuition:
Reduction:		$S \vdash_b q$ resp. $S \vdash_c q$ is true (wrt. all interpretations $I$ of $I'$ ) if $S \models_b q$ resp. $S \models_c q$ .
Proposition		
For a normal logic program $P$ and query $q$ we have that $(1)$ $M_{\rm rel}(-l)=(p,q)$ is consistent and each answer set contains $noAS$ iff $P \not\models_b q$ ; $M \mapsto M_{\rm rel}(-l)$		
and $M \cup M_{\rm rig}^{\rm D,\{\leftarrow q\}}$ is consistent and each answer set contains $noAS$ iff $P \models_{c} q$ .		
Trad C, (1'U Merva) HES-Programs Laby 4, 2007 1685	6 Red C (TV Verning HEX Ringsmin) Aby 4,2017 17/28	Foot C (10 Mercs) 162 Mingsmin ship4,5007 177 26

Query Answering: Language Extension	Query Answering: Language Extension Proposition	Query Answering: Language Extension
	Proposition	Proposition
Now we extend ASP with queries over subprograms:		
Definition A ground query atom is of form $S\vdash_{\Gamma} g,$ where $\ell\in\{b,c\}$ determines the type of the	For a logic program P with query atoms, the answer sets of P and $[P]$ , projected to the atoms in P, coincide.	For a logic program $P$ with query atoms, the answer sets of $P$ and $\{P_i,$ projected to the atoms in $P_i$ coincide.
is a normal logic (sub)program, and $q$ is a query over $S$ .	Example	Example
Ouery atoms may occur in bodies of ASP programs in place of ordinary atoms.  Intuition:	Suppose $P_{\rm gwer}$ guesses all edge selections in a graph and $P_{\rm cinet}$ derives $invalid$ if the current selection is not a Hamiltonian cycle.	Suppose $P_{paca}$ guesses all edge selections in a graph and $P_{deck}$ derives $imutid$ if the current selection is not a Hamiltonian cycle.
$S \vdash_b q$ resp. $S \vdash_c q$ is true (wrt. all interpretations $I$ of $P$ ) if $S \models_c q$ is $F$ expanding the semantics of such a program $P$ uses the following translation:		Then
$[P] = P_{ S^{r,q} \rightarrow modS_{S^{r,q}}} \cup \bigcup_{S^{r,d} \cap P} (M \cup M_{0g}^{S \cup \{r-2\}\} \in S^r}) \bigcup_{ I \rightarrow c_{S^r,pq}}$		(and thus $[P]$ ) has an answer set containing nor $P_{check} \vdash_c mvatial$ at hand does not contain a Hamiltonian group and the graph at hand does not contain a Hamiltonian cycle.
$\bigcup_{S \leftarrow a \text{ in } P} (M \cup M_{RR}^{N \cup \{ \leftarrow q \}}) _{a \rightarrow c_{Rr \rightarrow c}}$		Otherwise the program has at least one answer set but none of the answer sets
to Ab of		contains atom noHamiltonian.
Query Answering: Checking Conditions with Default-Negation	Outline	Discussion Alternative Approaches
	1 Motivation	■ Nested HEX-programs (Etter et al., 2013):
In general:	2 The Saturation Technique and its Bestrictions	Baesd on the ASP-extension of HEX-programs (with access to external
		sources) rather than plain ASP.
$\blacksquare$ Let program $P_{guess}$ span a search space of all objects to check.	Deciding Inconsistency of Normal Programs in Disjunctive ASP	
${\bf E}$ Let $P_{ched}$ check if the current guess satisfies the criteria and derive atom $ok$ in this case.	Query Answering over Subprograms	
Instead manually saturation whenever ok is true, just checks if ok is	5 Discussion	
instead mandaily saturating whenever on is true, just checks if on is		

Alternative Approaches    Nested HEX-programs [Eler et al., 2013]:   Based on the ASP-axtension of HEX-programs [Filer et al., 2013]:   Based on the ASP-axtension of HEX-programs [Filer et al., 2013]:   Based on the ASP-axtension of HEX-programs [Filer et al., 2013]:   Based on the ASP-axtension of HEX-programs [Filer et al., 2013]:   Based on the ASP-axtension of HEX-programs [Filer et al., 2013]:   Based on the ASP-axtension of HEX-programs [Filer et al., 2013]:   Based on the ASP-axtension of HEX-programs [Filer et al., 2013]:   Based on the ASP-axtension of HEX-programs [Filer et al., 2013]:   Based on the ASP-axtension of HEX-programs [Filer et al., 2013]:   Based on the ASP-axtension of HEX-programs [Filer et al., 2013]:   Based on the ASP-axtension of HEX-programs [Filer et al., 2013]:   Based on the ASP-axtension of HEX-programs [Filer et al., 2013]:   Based on the ASP-axtension of HEX-programs [Filer et al., 2013]:   Based on the ASP-axtension of HEX-programs [Filer et al., 2013]:   Based on the ASP-axtension of HEX-programs [Filer et al., 2013]:   Based on the ASP-axtension of HEX-programs [Filer et al., 2013]:   Based on the ASP-axtension of HEX-programs [Filer et al., 2013]:   Based on the ASP-axtension of HEX-programs [Filer et al., 2013]:   Based on the ASP-axtension of HEX-programs [Filer et al., 2013]:   Based on the ASP-axtension of HEX-programs [Filer et al., 2013]:   Based on the ASP-axtension of HEX-programs [Filer et al., 2013]:   Based on the ASP-axtension of HEX-programs [Filer et al., 2013]:   Based on the ASP-axtension of HEX-programs [Filer et al., 2013]:   Based on the ASP-axtension of HEX-programs [Filer et al., 2013]:   Based on the ASP-axtension of HEX-programs [Filer et al., 2013]:   Based on the ASP-axtension of HEX-programs [Filer et al., 2013]:   Based on the ASP-axtension of HEX-programs [Filer et al., 2013]:   Based on the ASP-axtension of HEX-programs [Filer et al., 2013]:   Based on the ASP-axtension of HEX-programs [Filer et al., 2013]:   Based on the ASP-axtension of HEX-	Eller et al., 2013]: Islon of HEX-programs (with access to external in ASP. In ASP. Songonents but no compliation into a single program. ornponents but no compliation into a single program. g weak constraints.	Alternative Approaches  Nesded HEX-programs [Eiter et al., 2013]: Based on the ASP-extension of HEX-programs (with access to external sources) rather than plain ASP.  Modular ASP [Tair et al., 2005]: Modular ASP [Tair et al., 2005]: Programs consisting of components but no compilation into a single program.  Manidol programs [Faber and Woltran, 2011]: Compilation-based using week constrains.  Stable-unstable semantics [Bogarats et al., 2016]: Similar goal (isolating the oracle program) and generalize to the PH, but no dedicated query atoms.
Approaches  X-programs [Eiler et al., 2013];  The ASP-extension of HEX-programs (with access to external uther than pain ASP.  Sonsisting of components but no compilation into a single program.	Elter et al., 2013]: sivin of HEX-programs (with access to external in ASP, 2005]: components but no compilation into a single program. er and Worlran, 2011]: g weak constraints.	(Eiter et al., 2013): In Stoin of HEX-programs (with access to eximitation of HEX-programs) (with access to eximal components but no compilation into a sing er and Wolfran, 2011): Its (Bogaerts et al., 2016): Its oracle program) and generalize to the Platoms.  Accessed
ati., 2013); HEX-programs (with access to external ents but no compilation into a single program.	rams (with access to ext o compilation into a single 11):	at., 2013): HEX-programs (with access to extend to but no compilation into a sing voltant 2011): aerts et at., 2016): program) and generalize to the Plance.
Teld (TO Service) restringuines sustaines sustaines feet (TO Service)	SELECTION SELECT	MGT (10'0'0') MGT-0pm asystom asystom and common and co
Phonence	овгада	Condusor
Discussion		Conclusion
Alternative Approaches		Two Restriction of Answer Set Programming
s (with access to external mplation into a single program.		<ul> <li>No meta-reasoning over collections of answer sets.</li> <li>Limitations of the saturation technique:         difficult to use for ASP laymen and restricted use of default-negation.</li> </ul>
eralize to the PH,		
■ Enroding is also related to debugging approaches (e.g. [Gebser et al., 2006, Oetsch et al., 2010]): Rather than explaining inconsistency we exploit it for query answering.		

#### Eiter, T. and Polleres, A. (2006). Towards automated integration of guess and check programs in answer set Bogaerts, B., Janhunen, T., and Tasharrofi, S. (2016). Stable-unstable semantics: Beyond NP with normal logic programs. programming: a meta-interpreter and applications. TPLP, 6(1-2):23-60. Eiter, T., Krennwallner, T., and Redl, C. (2013). HEX-programs with nested program calls. Ann. Math. Artif. Intell., 15(3-4):289-323. Eiter, T. and Gottlob, G. (1995). case. ■ Limitations of the saturation technique: difficult to use for ASP laymen and restricted use of default-negation. Implementation and application for nested HEX-program evaluation. A language extension of ASP program with dedicated query atoms. ■ Encoding for deciding inconsistency of a normal program. Encoding for query answering over a normal program. No meta-reasoning over collections of answer sets. Two Restriction of Answer Set Programming ■ More user-friendly alternative to saturation. Extension to non-ground queries. Contribution and Solution **Future Work** I Limitations of the saturation technique: difficult to use for ASP laymen and restricted use of default-negation. A language extension of ASP program with dedicated query atoms. ■ Encoding for deciding inconsistency of a normal program. ■ Encoding for query answering over a normal program. ■ No meta-reasoning over collections of answer sets. Two Restriction of Answer Set Programming More user-friendly alternative to saturation. Contribution and Solution Conclusion

# References II

in Logic Programming, Knowledge Representation, and Nonmonotonic Reasoning, volume 6565 of Lecture Notes in Computer Science, pages Faber, W. and Woltran, S. (2011).
Manifold answer-set programs and their applications. 44-63. Springer.

Gebser, M., Pührer, J., Schaub, T., and Tompits, H. (2008).
A meta-programming technique for debugging answer-set programs.

Gelfond, M. and Lifschitz, V. (1991).

Classical Negation in Logic Programs and Disjunctive Databases. New Generation Computing, 9(3-4):365-386

Oetsch, J., Pührer, J., and Tompits, H. (2010). Catching the ouroboros: On debugging non-ground answer-set programs.

July 4, 2017 26 / 26

### References III

In Vos. M. D. and Provetti, A., editors, *Answer Set Programming, Advances in Thery and Implementation, Proceedings of the 3rd Intl. ASP05 Workshop, Bath, UK, September 27-29, 2005,* volume 142 of *CEUR Workshop Proceedings, CEUR-WS.org*. Tari, L., Baral, C., and Anwar, S. (2005). A language for modular answer set programming: Application to ACC tournament scheduling.

On the computational cost of disjunctive logic programming: Propositional

Conference on Applications of Declarative Programming and Knowledge Management (INAP 2011), volume 7773 of LNAI, pages 1–10. Springer. In Tompits, H., editor, Proceedings of the Nineteenth International

July 4, 2017 24 / 26

July 4, 2017 23 / 26