$a_1 \vee \dots \vee a_n \leftarrow b_1, \dots, b_m,$ not b_{m+1}, \dots , not b_n , with classical literals a_i , and classical literals or an external atoms b_i . HEX-Programs HEX-programs extend ordinary ASP programs by external sources A HEX-program consists of rules of form Definition (HEX-programs) 1 Motivation Answer Set Programs and Extensions Explaining Inconsistency in

Christoph Redl

redl@kr.tuwien.ac.at



Vienna University of Technology TECHNISCHE UNIVERSITÄT WIEN VIEN VIEN VIENBUniversity of

July 6, 2017

3 Complexity Results 5 Envisaged Applic 4 Con 2

 $p \dots \text{external predicate name} \\ p \dots \text{external predicate name}$ q_i ... predicate names or constants t_j ... terms Definition (External Atoms) An external atom is of the form Semantics:

 $\begin{array}{l} 1+k+Fary \, \text{Boolean oracle function} \, f_{q_0};\\ \phi_P[q_1,\ldots,q_k](t_1,\ldots,t_l) \, \text{is true under assignment } \mathbf{A}\\ \text{iff} \, f_{kp}(\mathbf{A},q_1,\ldots,q_k,t_l,\ldots,t_l) = 1. \end{array}$

\$
Implementation
of &p

Reasoner

HEX-program

Motivation

Main idea

- Inconsistent programs are programs without answer sets.
 - Question: Why it such a program inconsistent?

Main idea

Motivation

- Inconsistent programs are programs without answer sets.
- Question: Why it such a program inconsistent?
- Previous work (e.g. on debugging):
 Focus on explaining why a particular interpretation is not an answer set.
 Focus on human+readable explanations (unsupported atoms, odd loops, etc).

Main idea

Motivation

- Inconsistent programs are programs without answer sets.
- Question: Why it such a program inconsistent?
- Previous work (e.g. on debugging):
 Focus on explaining why a particular interpretation is not an answer set.
 Focus on human-readable explanations (unsupported atoms, odd loops, etc).
 - Here: Focus on explanations which be easily expressed as constraints.

Mativation	Motivation
Motivation	Motivation
Mativation	Motivation

Main idea

- Inconsistent programs are programs without answer sets.
 - Question: Why it such a program inconsistent?
- Previous work (e.g. on debugging):
 Focus on explaining why a particular interpretation is not an answer set.
 Focus on human-readable explanations (unsupported atoms, odd loops, etc).
 - Here: Focus on explanations which be easily expressed as constraints.
- Envisaged application: improvements for programs with multiple components.

Contribution

- Characterizing of inconsistency in terms of atoms in the facts (EDB). ⇒ Notion of inconsistency reasons (IRs)

Motivation Motivation

Contribution

- Characterizing of inconsistency in terms of atoms in the facts (EDB).
 ⇒ Notion of inconsistency reasons (IRs)
 That is, we identify classes of instances which are inconsistent.

Contribution

Motivation

■ Characterizing of inconsistency in terms of atoms in the facts (EDB). ⇒ Notion of inconsistency reasons (IRs) That is, we identify classes of instances which are inconsistent.

■ Characterizing of inconsistency in terms of atoms in the facts (EDB).
 ⇒ Notion of inconsistency reasons (IRs)
 That is, we identify classes of instances which are inconsistent.

Contribution

■ Complexity results regarding the computation of IRs.

- Complexity results regarding the computation of IRs.
- A meta-programming technique for computing IRs for normal programs.

July 6, 2017 5/22

- Appropriate	Entition (Introductions of MCV) Reasons	Amount (L. M.) proceedings of the Co. Co.
Motivation	Outline	Formalizing Inconsistency Reasons
Contribution	Motivation	Basic idea
 ■ Characterizing of inconsistency in terms of atoms in the facts (EDB). ⇒ Notion of inconsistency reasons (IRs) That is, we identify classes of instances which are inconsistent. ■ Complexity results regarding the computation of IRs. ■ A meta-programming technique for computing IRs for normal programs. ■ A procedural algorithm for computing IRs for general programs. 	Explaining Inconsistency of HEX-Programs Complexity Results Computing Inconsistency Reasons Envisaged Application Conclusion	$lacksquare$ Consider programs P extended by atoms $I\subseteq \mathcal{D}$ from a given domain \mathcal{D} .
25/9 448.59/47 mm362.53±	Ello 100 Selec (10 Verse)	STA THE ANNUAL TOTAL STATE STA
Espaining inconsistery of HEX Programs	Epidaing troordering of HEX-Programs	Esplainty Incordatory of NEX-Programs
Esplaining incombining of HEX Programs	Explaining troonsistency of HEX-Programs	Equiting theorisisteray of MEX.Programs
Formalizing Inconsistency Reasons	Formalizing Inconsistency Reasons	Formalizing Inconsistency Reasons
		Formally:
Basic idea	Basic idea	Definition (Inconsistency Reason (IR))
$lacksquare$ Consider programs P extended by atoms $I\subseteq\mathcal{D}$ from a given domain \mathcal{D} .	$lacksquare$ Consider programs P extended by atoms $I\subseteq\mathcal{D}$ from a given domain \mathcal{D} .	Let P be a HEX-program and ${\mathcal D}$ be a domain of atoms.
■ Goal: Express reasons for inconsistency of $P \cup faax(I)$ in terms of I (where $facts(I) = \{a \leftarrow a \in I\}$), i.e., a ortlerion wrt. I which guarantees that $P \cup facts(I)$ is inconsistent.	■ Goal: Express reasons for inconsistency of $P \cup facts(I)$ in terms of I (where $facts(I) = \{a \leftarrow \mid a \in I\}$), (where $facts(I) = \{a \leftarrow \mid a \in I\}$), i.e., a criterion wrt. I which guarantees that $P \cup facts(I)$ is inconsistent.	An inconsistency reason (IR) of P wrt, \mathcal{D} is a pair $R=(R^+,R^-)$ of sets of atoms $R^+\subseteq \mathcal{D}$ and $R^-\subseteq \mathcal{D}$ with $R^+\cap I=\emptyset$ s.t. $P\cup facts(I)$ is inconsistent for all $I\subseteq \mathcal{D}$ with $R^+\subseteq I$ and $R^-\cap I=\emptyset$.
	■ Proposed notion: Use pairs of atoms from $\mathcal D$ which must occur resp. must not occur in I such that $P \cup facts(I)$ is inconsistent.	

July 6, 2017 8/22

Explaining the consistency of HEX-Programs	Explaining hoomisteery of HEX-Programs	Explaining translationary of HEX-Programs
Formalizing Inconsistency Reasons	Formal Results	Formal Results
Formally:	Proposition	Proposition
Definition (Inconsistency Reason (IR))	For all HEX-programs P and domains $\mathcal D$ such that $P \cup facts(I)$ is inconsistent for	For all HEX-programs P and domains ${\cal D}$ such that $P\cup facts(I)$ is inconsistent for
Let P be a HEX-program and ${\mathcal D}$ be a domain of atoms.	some set $I\subseteq \mathcal{D}$ of atoms, then there is an IR of P wrt. \mathcal{D} .	some set $I \subseteq \mathcal{D}$ of atoms, then there is an IR of P wrt. \mathcal{D} .
An inconsistency reason (IR) of P with, D is a pair $R=(R^+,R^-)$ of sets of atoms $R^+\subseteq D$ and $R^-\subseteq D$ with $R^+\cap R^-=\emptyset$ s.t. $P\cup facts(I)$ is inconsistent for all $I\subseteq D$ with $R^+\subseteq I$ and $R^-\cap I=\emptyset$.		Alternative characterization based on unfounded sets: Definition (Unfounded Set [Faber, 2005]) Given a HEX-program P and an assignment A, let U be any set of atoms
Example		appearing in P . Then, U is an <i>untounded set for P wit.</i> A it, for each rule $r \in P$ with $H(r) \cap U \neq \emptyset$, at least one of the following conditions holds:
Consider $P = \{\leftarrow a, not c; \ d \leftarrow b.\}$ and $\mathcal{D} = \{a,b,c\}.$		(i) some literal of $B(r)$ is false wrt. A ; or
An IR is $R=\{\{a\},\{c\}\}$ because $P\cup facts(I)$ is inconsistent for all $I\subseteq \mathcal{D}$ whenever $a\in I$ and $c\not\in I$.		(ii) some literal of $B(r)$ is false wrt. $A\setminus U$; or (iii) some atom of $H(r)\setminus U$ is true wrt. A .
rementation of the second of t	Cuttine Corporation (Continue)	Complexity Results
Proposition	Motivation	
Let P be a ground HEX-program and $\mathcal D$ be a domain. Then a pair of sets of atoms (R^+,R^-) with $R^+\subseteq \mathcal D$ $R^-\subseteq \mathcal D$ and $R^+\cap R^-=\emptyset$ is an IR of P iff for all classical models M of P either 0 (P $\subseteq M$ or 0) there is a nonempty unfounded set U of P wt. M such that $U\cap M\neq\emptyset$ and $U\cap (\mathcal D\setminus R^-)=\emptyset$.		Program cla
Relaxed proposition (sound but not complete for determining IRs):		Checking a subsert-initimal IR candidate $D_1^{h_1}c$ $D_2^{h_2}c$ Checking existence of a minimal IR $\Pi_2^{h_2}c$ $\Pi_2^{h_2}c$
Proposition	4 Computing Inconsistency Reasons	Table: Summary of Complexity Results
Let P be a ground HEX-program and \mathcal{D} be a domain such that $H(P) \cap \mathcal{D} = \emptyset$. For a pair of sets of atoms (R^+, R^-) with $R^+ \subseteq \mathcal{D}$ and $R^- \subseteq \mathcal{D}$, if for all classical models M of P we either have $(i) R^+ \subseteq M$ or $(i) R^- \cap M \neq \emptyset$ then (R^+, R^-) is an $H \circ QP$	5 Envisaged Application 6 Conclusion	

Computing Inconsistency Payasons				Compeny Hesults Computing Inconsistency Reasons		
Computing Propriet Seriory Peasons	Inconsistency Reasons for Normal ASP-Programs	A Meta-Program for Inconsistency Detection	For a normal ASP program P one can construct a positive disjunctive meta-program M^p with the following properties (e.g. [Eiter and Polleres, 2006]): $\blacksquare M^p$ is always consistent;	• If P is inconsistent, then M^P has a single answer set $A_{nw} = A(M^P)$ containing all atoms in M^P including a dedicated atom <i>incons</i> which does not appear in P , and	if P is consistent, then the answer sets of M ^P correspond one-to-one to those of P and none of them contains incons.	
Computing inconsistency Pagasons	Inconsistency Reasons for Normal ASP-Programs	A Meta-Program for Inconsistency Detection	For a normal ASP program P one can construct a positive disjunctive mata-program M^P with the following properties (e.g. [Eiter and Polleres, 2006]): $\blacksquare M^P$ is always consistent:	$ \begin{tabular}{ll} \hline & if P is inconsistent, then M^P has a single answer set $A_{\rm nor} = A(M^P)$ containing all atoms in M^P including a dedicated atom $incons$ which does not appear in P and P and P.$	if P is consistent, then the answer sets of M ^P correspond one-to-one to those of P and none of them contains incons.	Then, the atom $incons$ in the answer set(s) of M^{p} represents inconsistency of the original program P .

	Inconsiste
automora farinare and farinare	onsistency Beasons for Normal ASP-Programs
	-

A Meta-Program for IR Computation	
IRs of P can then be computed using the meta-program: $\tau(\mathcal{D},P)=M^P$	£
$\cup \{a^+ \vee a^- \vee a^r \mid a \in \mathcal{D}\} \ \cup$	(2)
$\cup \left\{ a \leftarrow a^+; \leftarrow a, a^-; \ a \lor \bar{a} \leftarrow a^{\chi}; \ \ a \in \mathcal{D} \right\}$	(3)
$\cup \left\{ \leftarrow \text{ not } incoms \right\},$ where $a^+ = a^-$ and \bar{a} are near stone for all stone $a \in \mathcal{D}$	4)

Proposition Let P be an ordinary normal program and D be a domain. Then (R^+,R^-) is an IR of P wrt. D iff $\tau(D,P)$ has an answer set $A \supseteq \{d'' \mid \sigma \in \{+,-\}, a \in R''\}$.

tency Reasons for General HEX-Programs Inconsis

 \blacksquare Deciding if a general HEX-program has an IR is Σ_i^{γ} -complete but reasoning tasks over a general HEX-program are only on the second level of the polynomial hierarchy.

Challenges

Inconsistency Reasons for General HEX-Programs

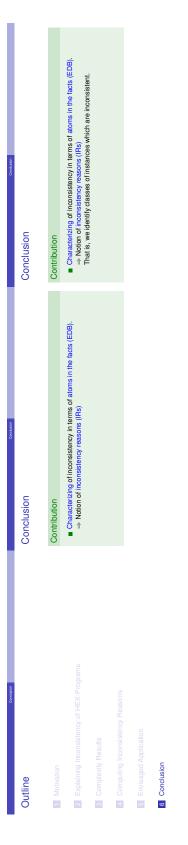
Challenges

- \blacksquare Deciding if a general HEX-program has an IR is Σ_3^{γ} -complete but reasoning tasks over a general HEX-program are only on the second level of the polynomial hierarchy.
- $\blacksquare \Rightarrow \mathsf{Computing} \text{ the IRs of a general HEX-program cannot be polynomially } \mathsf{reduced} \text{ to a meta-HEX-program (unless } \Sigma_{j}^{g} = \Sigma_{j}^{g}).$

5 Envisaged Application 3 Complexity Results 6 Conclusion 4 Comp \blacksquare Deciding if a general HEX-program has an IR is Σ_3^{γ} -complete but reasoning tasks over a general HEX-program are only on the second level of the polynomial hierarchy. $\blacksquare \Rightarrow$ Computing the IRs of a general HEX-program cannot be polynomially reduced to a meta-HEX-program (unless $\Sigma_3^2=\Sigma_2^2$). Giving up completeness For HEX-programs P without disjunctions we can specify an encoding for computing its IRs, which is sound but not complete. Inconsistency Reasons for General HEX-Programs Using a procedural algorithm ■ Remedies: Challenges \blacksquare Deciding it a general HEX-program has an IR is Σ_3^{+} complete but reasoning tasks over a general HEX-program are only on the second level of the polynomial hierarchy. $\blacksquare \Rightarrow$ Computing the IRs of a general HEX-program cannot be polynomially reduced to a meta-HEX-program (unless $\Sigma_1^2=\Sigma_2^2)$. Giving up completeness For HEX-programs P without disjunctions we can specify an encoding for computing its IRs, which is sound but not complete. Inconsistency Reasons for General HEX-Programs ■ Remedies: Challenges

Per kapyal Applanton	Learning Over Multiple Program Components HEX-program evaluation is based on program splitting.	 ■ We want to form a committee of employees. ■ Some pairs of persons have conflicts of interests. ■ The competences of the committee depend on the persons involved. ■ Competences can depend nonmonotonically on the members. ■ Constraints define the competences the committee should have. P = {n: in(X) \ vou(X) \rightarrow person(X). r₂ : ← in(X), in(Y), conflict(X, Y). r₃ : comp(X) ← & competences [in](X). r₄ : ← not comp(technical), not comp(financial).}
Envanged Aptication	Learning Over Multiple Program Components HEX-program evaluation is based on program splitting. Example	 ■ We want to form a committee of employees. ■ Some pairs of persons have conflicts of interests. ■ The competences of the committee depend on the persons involved. ■ Competences can depend nonmonotonically on the members. ■ Constraints define the competences the committee should have.
Evinabel Asptanton	Learning Over Multiple Program Components HEX-program evaluation is based on program splitting.	

Main Idea: Associate an IR R of a later program component with a constraint c_R which we propagate to predecessors. inconsistency reason R'----- detect inconsistency Learning Over Multiple Program Components $P_2 = \{r_3, r_4\}$ add answer set as input atoms $P_1 = \{r_1, r_2\}$ add as constraint c_R **Main Idea:** Associate an IR R of a later program component with a constraint c_R which we propagate to predecessors. Learning Over Multiple Program Components ■ The competences of the committee depend on the persons involved. ■ Competences can depend normonotonically on the members. ■ Constraints define the competences the committee should have. Figure: Evaluation graph of the program from the previous example r_4 : \leftarrow not comp(technical), not comp(financial).} Learning Over Multiple Program Components HEX-program evaluation is based on program splitting. $r_3: comp(X) \leftarrow \&competences[in](X).$ Some pairs of persons have conflicts of interests. $r_2:\ \leftarrow in(X), in(Y), conflict(X,Y).$ $P = \{r_1 : in(X) \lor out(X) \leftarrow person(X).$ We want to form a committee of employees. $P_1 = \{r_1, r_2\}$



Condusion	Conclusion
Conclusion	clusion
Condusion	onclusion

Contribution

- Characterizing of inconsistency in terms of atoms in the facts (EDB).

 Notice of inconsistency reasons (IRS)

 That is, we identify classes of instances which are inconsistent.

 Complexity results regarding the computation of IRs.

Contribution

- Characterizing of inconsistency in terms of atoms in the facts (EDB).

 ⇒ Notion of inconsistency reasons (IRs).

 That is, we identify classes of instances which are inconsistent.

 Competity results regarding the computation of IRs.

 A meta-programming technique for computing IRs for normal programs.

Contribution

- Characterizing of inconsistency in terms of atoms in the facts (EDB).
 ⇒ Notion of inconsistency reasons (IRs)
 That is, we identify classes of instances which are inconsistent.
 Complexity results regarding the computation of IRs.
 A mate-programming technique for computing IRs for normal programs.
 A procedural algorithm for computing IRs for general programs.

Conclusion

Contribution

- Characterizing of inconsistency in terms of atoms in the facts (EDB).
- Notion of inconsistency reasons (IRs)
 That is, we identify classes of instances which are inconsistent.

 Complexity results regarding the computation of IRs.

 A meta-programming technique for computing IRs for normal programs.

Ongoing Work

Application of the results for algorithmic improvements.

References I

- Eiter, T. and Polleres, A. (2006). Towards automated integration of guess and check programs in answer set programming: a meta-interpreter and applications.
- Faber, W (2005).
 Unfounded sets for disjunctive logic programs with arbitrary aggregates.
 Unfounded sets for disjunctive logic programs with arbitrary aggregates in Proceedings of the Eighth International Conference on Logic Programming and Normonotonic Reasoning (LPNMR 2005), Diamante, Italy, September 5-8, 2005, volume 3865, pages 40–92. Springer.

July 6, 2017 22 / 22

July 6, 2017 21/22