A Sequent Calculus for Answer Set Entailment (Extended Abstract)

Thomas Eiter and **Tobias Geibinger**
Knowledge-based Systems Group, Institute of Logic and Computation, TU Wien, Austria
{thomas.eiter, tobias.geibinger}@tuwien.ac.at

1 Background

Answer Set Programming (ASP) is a symbolic rule-based reasoning formalism that has been used for various AI applications in numerous domains.ASP allows for a declarative encoding of problems in a succinct manner. Solutions for them are obtained from *answer sets*, which result from the evaluation of the encoding using an ASP solver. In this work, we study answer set entailment, which intuitively formalises which sentences hold in every answer set of a program.

Given the practical usage of ASP, questions of explainability have been raised. Those questions or problems generally concern themselves with either answering why certain literals hold in a particular answer set or why there is no answer set at all, cf. Fandinno and Schulz (2019) for a survey. Both these questions can be formulated through answer set entailment. However, simply stating a valid entailment is in general not an explanation on its own and needs to be justified. Such justifications can be obtained via a formal proof system.

For such a system of inference rules, i.e., proof system, to be meaningful for explainability, we argue that it should adhere to the following informal requirements: (R1) the rules and axioms of the proof system should be, like classical proof calculi, local and encode simple semantic concepts, (R2) the proof system should support commonly used language features of ASP, and (R3) the proofs should be concise and interpretable. Of course, those requirements are open to interpretation, but they are nonetheless useful desiderata.

Although ASP has some efficient solvers available (Leone et al. 2006; Gebser et al. 2019; Alviano et al. 2013) and its model-theoretic properties have been studied extensively, the same cannot be said for proof-theoretic investigations.

Given the nonmonotonic nature of ASP, obtaining a proof calculus which fulfils all those criteria is non-trivial, but requirements (R1-R3) are important if one really wants to characterise the semantics in a way that can serve the explainability of the formalism.

In this work, we introduce a sound and complete sequent proof calculus for entailment in equilibrium logic (EL). The latter generalises ASP to arbitrary propositional theories and serves as a theoretical foundation of stable model semantics. It is obtained by imposing certain conditions of the logic of here and there (HT).

The inference relation we axiomatise is the following.

Definition 1 (Equilibrium Entailment)**.** *Given theories* Γ *and* Δ*, we say that* Γ equilibrium entails Δ*, written* $\Gamma \approx \Delta$*, if for every equilibrium model* I *of* Γ*,* $I \models \varphi$ *for some* $\varphi \in \Delta$*.*

We argue that equilibrium entailment is a useful concept w.r.t. justification and explainability. For example, $\Gamma \approx \bot$ indicates that Γ is infeasible. By providing a proof of said inference, one effectively justifies that Γ and, given that the rules of the proof system are simple and interpretable, provides a baseline as to how an explanation should proceed. This of course also holds in the general case when one wants to know why a particular literal or sentence holds in every equilibrium model. Furthermore, equilibrium entailment is also applicable when one already has a particular equilibrium model I and seeks to explain why certain atoms are, respectively, are not, in the model.

Our calculus is very close to the original sequent calculus for classical logic and also takes inspiration from sequent calculi that were defined so related, but distinct, nonmonotonic formalisms (Olivetti 1992; Bonatti and Olivetti 1997). In particular, we similarly use an anti-sequent calculus, which is a calculus that axiomatises non-entailment.

Arguably, our approach is to be more aligned with the requirements (R1-R3) from above as previous calculi. The generated sequent proofs are, given some knowledge of proof theory, easy to parse and do not require multiple stages, auxiliary concepts, or complicated side conditions.

Our main contributions are as follows:

- We provide an axiomatisation of answer set entailment in the form of a sequent calculus.
- We discuss an alternative characterisation that does not use the anti-sequent calculus, but requires rules which can only be applied in certain language fragments.
- We investigate how the calculus behaves when restricted to ASP programs and show which rules remain necessary.
- We show that answer set entailment is a useful framework for explainability and our calculus strengthens this by providing a formal way to justify entailment.

2 Sequent Calculus

We now introduce our notion of sequent, which will be the main building block for our proof system.

Definition 2. *An* equilibrium sequent *is of the form* $\Gamma \mathrel{|\!\!\sim} \Delta$*, where* Γ *and* Δ *are both sets of formulas.*

$$\frac{\Gamma, \varphi, \psi \mid\!\sim \Delta}{\Gamma, \varphi \land \psi \mid\!\sim \Delta} (\land_l) \qquad \frac{\Gamma \mid\!\sim \Delta, \varphi \qquad \Gamma \mid\!\sim \Delta, \psi}{\Gamma \mid\!\sim \Delta, \varphi \land \psi} (\land_r) \qquad \frac{\Gamma \mid\!\sim \Delta}{\Gamma \mid\!\sim \Delta, \varphi} (w_r) \qquad \frac{\Gamma \mid\!\sim \Delta, \bot}{\Gamma \mid\!\sim \Delta} (\bot)$$

$$\frac{\Gamma \mid\!\sim \Delta, \varphi, \psi}{\Gamma \mid\!\sim \Delta, \varphi \lor \psi} (\lor_r) \qquad \frac{\Gamma, \varphi \mid\!\sim \Delta \qquad \Gamma, \psi \mid\!\sim \Delta, \varphi}{\Gamma, \varphi \lor \psi \mid\!\sim \Delta} (\lor_l) \qquad \frac{\Gamma \mid\!\sim \Delta, \varphi}{\Gamma, \neg\varphi \mid\!\sim \Delta} (\neg_l)$$

$$\frac{\Gamma, \varphi \mid\!\sim \Delta, \psi}{\Gamma \mid\!\sim \Delta, \varphi \supset \psi} (\supset_r) \qquad \frac{\Gamma, \psi \mid\!\sim \Delta, \neg\varphi \qquad \Gamma \mid\!\sim \Delta, \varphi}{\Gamma, \varphi \supset \psi \mid\!\sim \Delta} (\supset_l) \qquad \frac{\Gamma, \varphi \mid\!\sim \Delta}{\Gamma \mid\!\sim \Delta, \neg\varphi} (\neg_r)$$

$$\frac{\Gamma \mid\!\sim \Delta, \varphi \qquad \Gamma, \varphi \mid\!\sim \Sigma}{\Gamma \mid\!\sim \Delta, \Sigma} (RCut) \qquad \frac{\Gamma \mid\!\sim \bigwedge S \cup \neg\overline{S} \qquad \Gamma, \neg\neg S, \neg\overline{S} \dashv_{HT} \bigwedge_{p \in S} p \lor \neg p}{\Gamma \mid\!\sim \Delta} (\dashv) \text{ where } S \subseteq \mathit{Var}(\Gamma)$$

Figure 1: Rules of the sequent calculus

An equilibrium sequent $\Gamma \mid\!\sim \Delta$ *is* satisfied, *if* $\Gamma \approx \Delta$.

Our calculus has two axioms which will also be the allowed initial sequents in our derivations.

Definition 3. *An* initial sequent *is either*

(i) $\Gamma \mid\!\sim \Delta, \neg p$ *if* Γ *is a set of atoms and* $p \notin \Gamma$*, or*

(ii) $\Gamma, \varphi \mid\!\sim \Delta, \varphi$.

The rules of the calculus are shown in Figure 1. In the ($\dashv$)-rule the right premise uses a refutation calculus for the logic of there and there (HT). Intuitively, it states that there is some HT model of the left-hand side which does not satisfy the right-hand side of the anti-sequent. Furthermore, $\overline{S}$ refers to the complement of S.

Now, as usual, a sequent proof is defined as follows.

Definition 4. *A* derivation *of a sequent* $\Gamma \mid\!\sim \Delta$ *is a tree that is rooted in the sequent, the leaves are* initial sequents *and parent nodes are generated by the application of some rule.*

The initial sequents, as well as the rules, are all sound.

Proposition 1. *The initial sequents given in Definition 3 and the rules given in Figure 1 are all sound, i.e., if the sequents in the premise are satisfied, then so is the sequent in the conclusion.*

The proof proceeds by showing the statement for each rule which together with the initial sequents constitute our sequent calculus ELK.

Theorem 1. *The sequent calculus* ELK *is sound, i.e., if* $\Gamma \mid\!\sim \Delta$ *is derivable in* ELK*, then* $\Gamma \approx \Delta$ *holds.*

We note that the rules of ELK are not invertible. However, using the *RCut* rule as and the usual counter model construction, we can show the following.

Theorem 2. *If* $\Gamma \approx \varphi$*, then* $\Gamma \mid\!\sim \varphi$ *is derivable in* ELK.

The sequent calculus ELK thus satisfies the following.

Corollary 1 (Main Result)**.** *The sequent calculus* ELK *is sound and complete.*

3 Discussion

In the paper, we have introduced an axiomatisation of answer set entailment in the form of a sequent calculus for equilibrium logic. The calculus was then shown to be sound as well as complete. All rules, except one are natural and structurally simple, the remaining rule utilises an refutation calculus.

We argue that equilibrium entailment provides a useful framework for explanations in equilibrium logic and thus ASP, and our calculus gives insight into why such an entailment is valid. We do not claim that proofs in the sequent calculus are already accessible explanations, but as we discuss in the paper, they serve as an overarching theoretical framework from which explanations can be derived.

Both the calculus in itself, as well as the potential usage in explainability clearly make our paper relevant for KR. Due to it being a generalisation of ASP, equilibrium logic is of great interest to the community and proof theoretic investigations of it are sparse. Our work thus helps to fill this gap.

References

Alviano, M.; Dodaro, C.; Faber, W.; Leone, N.; and Ricca, F. 2013. WASP: A native ASP solver based on constraint learning. In *Proceedings of the 12th International Conference on Logic Programming and Nonmonotonic Reasoning (LPNMR 2013)*, volume 8148 of *LNCS*, 54–66. Springer.

Bonatti, P. A., and Olivetti, N. 1997. A sequent calculus for circumscription. In *Selected Papers of the 11th International Computer Science Logic Workshop (CSL 1997)*, volume 1414 of *LNCS*, 98–114. Springer.

Fandinno, J., and Schulz, C. 2019. Answering the "why" in answer set programming - A survey of explanation approaches. *Theory and Practice of Logic Programming* 19(2):114–203.

Gebser, M.; Kaminski, R.; Kaufmann, B.; and Schaub, T. 2019. Multi-shot ASP solving with clingo. *Theory and Practice of Logic Programming* 19(1):27–82.

Leone, N.; Pfeifer, G.; Faber, W.; Eiter, T.; Gottlob, G.; Perri, S.; and Scarcello, F. 2006. The DLV system for knowledge representation and reasoning. *ACM Transactions on Computational Logic* 7(3):499–562.

Olivetti, N. 1992. Tableaux and sequent calculus for minimal entailment. *Journal of Automated Reasoning* 9(1):99–139.