A Sequent Calculus for Answer Set Entailment

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

Abstract

Answer Set Programming (ASP) is a popular nonmonotonic formalism used for common-sense reasoning and problem-solving based on stable model semantics. Equilibrium logic is a generalisation of ASP for arbitrary propositional theories and thus provides a logical characterisation of the nonmonotonic stable model semantics. In difference to classical logic, which can be defined via proof or model theory, nonmonotonic reasoning formalisms are defined via their models exclusively. Equilibrium logic is no exception here, as it has no proper proof-theoretic axiomatisation. Besides this being a theoretical imbalance, it also has consequences regarding notions of justification and explainability. In this work, we fill this gap by providing a sequent calculus for answer set entailment. Our calculus builds upon ideas from existing calculi for other nonmonotonic formalisms and utilises calculi for the logic of here and there, which is the underlying base logic of equilibrium logic. We show that the calculus is sound and complete and discuss pitfalls as well as alternative axiomatisations. Finally, we address how our approach can be of use for explainability in ASP.

1 Introduction

Answer Set Programming (ASP) is a symbolic rule-based reasoning formalism that has been used for various AI applications in numerous domains [Erdem *et al.*, 2016; Falkner *et al.*, 2018], among them scheduling [Ali *et al.*, 2023; Abels *et al.*, 2019; Yli-Jyrä *et al.*, 2023], product configuration [Comploi-Taupe *et al.*, 2022], life sciences [Erdem and Oztok, 2015], health insurance [Beierle *et al.*, 2005], or psychology [Inclezan, 2015], to mention a few. 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. This concept itself is already of practical interest, and has been applied for diagnosis [Eiter *et al.*, 1999] and planning [Eiter *et al.*, 1997]. Notably, answer set entailment is at the core of abductive explanations [Eiter *et al.*, 1997].

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, incorporating the logical base of ASP in its inference rules.

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 for our purposes.

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. The few proof systems that do exist for answer set entailment [Gebser and Schaub, 2013; Bonatti *et al.*, 2008; Pearce *et al.*, 2000] are arguably weak on at least one of the requirements each.

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. Our calculus is very close to the original sequent calculus for classical logic invented by Gentzen [1935] and also takes inspiration from sequent calculi that were defined so related, but distinct, nonmonotonic formalisms [Olivetti, 1992; Bonatti and Olivetti, 1997a; Bonatti and Olivetti, 1997b]. In particular, we similarly use an anti-sequent calculus, which is a calculus that axiomatises non-entailment. As we will argue, without this component, even simple entailments may not be provable and that by adding a single non-entailment rule, which encodes the nonmonotonic stability condition inherent in ASP, we achieve a complete characterisation of answer set 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.

Briefly summarised, 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.

Our work enriches the rather sparse set of proof systems for ASP and equilibrium logic. Furthermore, the provided proofs are interpretable and can be translated into natural language as a basis for human-understandable explanations.

Missing proofs of the results are included in an extended version.

2 Background

Propositional *equilibrium logic (EL)* [Pearce, 1996; Pearce, 2006] generalises ground ASP semantics to arbitrary propositional formulas i.e. no restriction to rules.

We assume a denumerable set of propositional variables At. Formulas of EL are then defined in the usual manner over logical connectives $\wedge, \vee, \supset, \neg$ and logical constant $\bot$.

The semantics of EL is based on the *logic of here and there* (also called *HT logic*), which is defined over the same language. An interpretation in HT logic is a pair $\langle H, T\rangle$ where $H \subseteq T \subseteq At$. Whether an HT interpretation $\langle H, T\rangle$ is a *model* of a formula φ (denoted by $\langle H, T\rangle \models \varphi$) is inductively defined as follows: $\langle H, T\rangle \not\models \bot$, $\langle H, T\rangle \models p$ iff $p \in H$; $\langle H, T\rangle \models \varphi \wedge \psi$ iff $\langle H, T\rangle \models \varphi$ and $\langle H, T\rangle \models \psi$; $\langle H, T\rangle \models \varphi \vee \psi$ iff $\langle H, T\rangle \models \varphi$ or $\langle H, T\rangle \models \psi$; $\langle H, T\rangle \models \varphi \supset \psi$ iff $T \models \varphi \supset \psi$, and $\langle H, T\rangle \not\models \varphi$ or $\langle H, T\rangle \models \psi$; $\langle H, T\rangle \models \neg\varphi$ iff $T \not\models \varphi$ and $\langle H, T\rangle \not\models \varphi$.

Note that whenever $H = T$, $\langle H, T\rangle \models \varphi$ corresponds to the satisfaction relation of classical propositional logic. By slight abuse of notation, we use $T \models \varphi$ to refer to $\langle T, T\rangle \models \varphi$ and call such models *classical* or *total*.

From the above semantics, two useful observations follow. The first is that $\langle H, T\rangle \models \varphi$ implies $\langle T, T\rangle \models \varphi$. This property is also called *persistence*. The other important property is that $\langle H, T\rangle \models \neg\varphi$ iff $T \models \neg\varphi$, i.e., negation is only evaluated over T.

An HT model $\langle H, T\rangle$ of a formula φ is an *equilibrium model* iff $H = T$ and for any other HT interpretation $\langle H', T\rangle$ such that $H' \subset H$, $\langle H', T\rangle \not\models \varphi$. The latter conditions is also referred to as *stability*. We denote an equilibrium model $\langle T, T\rangle$ by T whenever convenient and we use a single set I to refer to classical interpretations. As usual, a set of formulas is called a *theory*, and we say that an interpretation is a model of a theory if it satisfies all contained formulas.

ASP programs are generally sets of rules of the form $H \leftarrow B$, where H and B are sets of literals. In EL, such a rule is encoded as an implication $\bigwedge B \supset \bigvee H$ and a program is then a theory consisting such an implication for each rule. As has been shown by Pearce [1996], the equilibrium models of such an encoded program amount to its answer sets.

From now on, whenever we talk about a *program*, we refer to a theory which encodes an ASP program as described above.

We use $\Gamma \models \varphi$ to denote that a theory classically entails φ and $\Gamma \models_{HT} \varphi$ for entailment in HT.

Furthermore, by $Var(\Gamma)$ we denote the set of atoms appearing in Γ and given a set S of atoms, we define $\neg S = \{\neg p \mid p \in S\}$ and $\neg\neg S = \{\neg\neg p \mid p \in S\}$. Whenever Γ is clear from context, $\overline{S} = Var(\Gamma) \setminus S$ denotes the *complement* of S.

Sequent calculus was invented by Gentzen [1935] for intuitionistic as well as classical propositional and predicate logic. It performs syntactic operations on pairs of sequences of rules creating proof trees. We will make use of a so called anti-sequent calculus for HT. This calculus, introduced by Oetsch and Tompits [2011] is a refutation calculus and thus axiomatises non-entailment. An anti-sequent $\Gamma \dashv_{HT} \Delta$ consists of sets of formulas Γ and Δ. Figure 1 shows the rules of the anti-sequent calculus for HT; they are slightly reformulated but equivalent to the original calculus. An anti-sequent $\Gamma \dashv_{HT} \Delta$ is an initial sequent if Γ and Δ are disjoint sets of literals and Γ is consistent, i.e., does not contain both an atom and its negation. $\Gamma \dashv_{HT} \Delta$ can be derived with the above rules and axioms iff $\Gamma \not\models_{HT} \Delta$ holds.

3 Equilibrium Entailment

The inference relation we seek to 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* $\varphi \in \Delta$*.*

This notion is slightly different from the equilibrium entailment defined by Pearce [2006] but more aligned with answer set entailment. In the definition of Pearce, equilibrium entailment falls back to classical entailment whenever Γ has no stable models, which is not suitable for our intentions as we would like $\Gamma \approx \bot$ whenever Γ has no equilibrium models.

We argue that equilibrium entailment is a useful concept w.r.t. justification and explainability. We have already mention the special case when $\Gamma \approx \bot$ 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.

To apply entailment here, one needs to handle the nondeterminism inherent in stable model semantics. This motivates the next definition, which is inspired by a similar concept used for ASP justification graphs [Pontelli *et al.*, 2009] and

$$\frac{\Gamma \dashv_{HT} \Delta, \neg\varphi}{\Gamma, \neg\neg\varphi \dashv_{HT} \Delta}\ (\neg\neg_l) \qquad \frac{\Gamma, \neg\varphi \dashv_{HT} \Delta}{\Gamma \dashv_{HT} \Delta, \neg\neg\varphi}\ (\neg\neg_r) \qquad \frac{\Gamma, \varphi, \psi \dashv_{HT} \Delta}{\Gamma, \varphi \wedge \psi \dashv_{HT} \Delta}\ (\wedge_l) \qquad \frac{\Gamma \dashv_{HT} \Delta, \varphi}{\Gamma \dashv_{HT} \Delta, \varphi \wedge \psi}\ (\wedge_{r_1})$$

$$\frac{\Gamma \dashv_{HT} \Delta, \psi}{\Gamma \dashv_{HT} \Delta, \varphi \wedge \psi}\ (\wedge_{r_2}) \qquad \frac{\Gamma \dashv_{HT} \Delta, \varphi, \psi}{\Gamma \dashv_{HT} \Delta, \varphi \vee \psi}\ (\vee_r) \qquad \frac{\Gamma, \varphi \dashv_{HT} \Delta}{\Gamma, \varphi \vee \psi \dashv_{HT} \Delta}\ (\vee_{l_1}) \qquad \frac{\Gamma, \psi \dashv_{HT} \Delta}{\Gamma, \varphi \vee \psi \dashv_{HT} \Delta}\ (\vee_{l_2})$$

$$\frac{\Gamma \dashv_{HT} \Delta, \varphi, \neg\psi}{\Gamma, \varphi \supset \psi \dashv_{HT} \Delta}\ (\supset_{l_1}) \qquad \frac{\Gamma, \psi \dashv_{HT} \Delta}{\Gamma, \varphi \supset \psi \dashv_{HT} \Delta}\ (\supset_{l_2}) \qquad \frac{\Gamma, \neg\varphi \dashv_{HT} \Delta}{\Gamma, \varphi \supset \psi \dashv_{HT} \Delta}\ (\supset_{l_3}) \qquad \frac{\Gamma, \varphi \dashv_{HT} \Delta, \psi}{\Gamma \dashv_{HT} \Delta, \varphi \supset \psi}\ (\supset_{r_1})$$

$$\frac{\Gamma, \neg\psi \dashv_{HT} \Delta, \neg\varphi}{\Gamma \dashv_{HT} \Delta, \varphi \supset \psi}\ (\supset_{r_2}) \qquad \frac{\Gamma, \neg\psi \dashv_{HT} \Delta, \neg\varphi}{\Gamma, \neg(\varphi \supset \psi) \dashv_{HT} \Delta}\ (\neg\supset_l) \qquad \frac{\Gamma, \neg\varphi \dashv_{HT} \Delta}{\Gamma \dashv_{HT} \Delta, \neg(\varphi \supset \psi)}\ (\neg\supset_{r_1})$$

$$\frac{\Gamma \dashv_{HT} \Delta, \neg\psi}{\Gamma \dashv_{HT} \Delta, \neg(\varphi \supset \psi)}\ (\neg\supset_{r_2}) \qquad \frac{\Gamma, \neg\varphi \dashv_{HT} \Delta}{\Gamma, \neg(\varphi \wedge \psi) \dashv_{HT} \Delta}\ (\neg\wedge_{l_1}) \qquad \frac{\Gamma, \neg\psi \dashv_{HT} \Delta}{\Gamma, \neg(\varphi \wedge \psi) \dashv_{HT} \Delta}\ (\neg\wedge_{l_2})$$

$$\frac{\Gamma \dashv_{HT} \Delta, \neg\varphi, \neg\psi}{\Gamma \dashv_{HT} \Delta, \neg(\varphi \wedge \psi)}\ (\neg\wedge_r) \qquad \frac{\Gamma, \neg\varphi, \neg\psi \dashv_{HT} \Delta}{\Gamma, \neg(\varphi \vee \psi) \dashv_{HT} \Delta}\ (\neg\vee_l) \qquad \frac{\Gamma \dashv_{HT} \Delta, \neg\varphi}{\Gamma \dashv_{HT} \Delta, \neg(\varphi \vee \psi)}\ (\neg\vee_{r_1})$$

$$\frac{\Gamma \dashv_{HT} \Delta, \neg\psi}{\Gamma \dashv_{HT} \Delta, \neg(\varphi \vee \psi)}\ (\neg\vee_{r_2})$$

Figure 1: Rules of the HT anti-sequent calculus

also Pearce's fix-point characterisation of equilibrium models [Pearce, 2006].

Definition 2 (Assumption Set)**.** *Given an interpretation I, a set of literals A is an* assumption set *w.r.t I if $I \models \bigwedge \neg A$.*

An assumption set is minimal *whenever it is a $\subseteq$-minimal assumption set w.r.t I.*

The intuition behind assumption sets will be more obvious after the following proposition.

Proposition 1. *Given a theory Γ with equilibrium model I and some $p \in Var(\Gamma)$, then there is a minimal assumption set A such that (i) $\Gamma, \neg A \approx p$ if $p \in I$, and (ii) $\Gamma, \neg A \approx \neg p$ if $p \notin I$.*

The basic idea is that the assumption set fixes the truth values of a necessary but minimal set of atoms to the truth value they have in the given model. Positive assumptions have to be added using double negation due to nonmonotonicity.

Example 1. *Consider the theory consisting of the single formula $a \vee b$. This theory has two equilibrium models $\{a\}$ and $\{b\}$. For consider the former and suppose we want to answer why a is in the model. The minimal assumption sets for $\{a\}$ are $\{\neg a\}$ and $\{b\}$ and clearly $\Gamma, \neg\neg a \approx a$ and $\Gamma, \neg b \approx a$ both hold. Intuitively, the assumption $\neg\neg a$ expresses that a is assumed to be true, whereas $\neg b$ assumes that b is false.*

Note that in the above example, simply assuming the truth of the atom we sought to justify was valid. It could be argued that for a proper explanation, only the latter, assuming b false, is preferable. However, we will leave such explanatory preferences for a different work.

We note the following observation, which is in the folklore and can be easily shown.

Proposition 2. $\Gamma \approx \neg p$ *if $p \notin Var(\Gamma)$.*

4 Sequent Calculus

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

Definition 3. *An* equilibrium sequent *is of the form $\Gamma \mathrel{|\!\sim} \Delta$, where Γ and Δ are both sets of formulas.*

An equilibrium sequent $\Gamma \mathrel{|\!\sim} \Delta$ is satisfied*, if $\Gamma \approx \Delta$.*

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

Definition 4. *An* initial sequent *is either*

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

(ii) $\Gamma, \varphi \mathrel{|\!\sim} \Delta, \varphi$.

Those initial sequents hold for any Γ and φ which follows from the Proposition 2 and from the fact that every equilibrium model is a classical model.

Proposition 3. *The initial sequents given in Definition 4 are sound.*

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

Definition 5. *A* derivation *of a sequent $\Gamma \mathrel{|\!\sim} \Delta$ 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 basic rules, are all sound.

Proposition 4. *The basic rules given in Figure 2 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.

Figure 2 shows the basic rules of our sequent calculus, which concern handling the logical connectives and weakening on the right. Note that most rules resemble their classical counterparts, except for $(\vee_l)$ and $(\supset_l)$. The former allows us to directly exclude disjuncts which would not be part of an equilibrium model as the following example shows.

$$\frac{\Gamma, \varphi, \psi \mathrel{|\!\sim} \Delta}{\Gamma, \varphi \wedge \psi \mathrel{|\!\sim} \Delta}\ (\wedge_l) \qquad \frac{\Gamma \mathrel{|\!\sim} \Delta, \varphi \qquad \Gamma \mathrel{|\!\sim} \Delta, \psi}{\Gamma \mathrel{|\!\sim} \Delta, \varphi \wedge \psi}\ (\wedge_r) \qquad \frac{\Gamma \mathrel{|\!\sim} \Delta}{\Gamma \mathrel{|\!\sim} \Delta, \varphi}\ (w_r) \qquad \frac{\Gamma \mathrel{|\!\sim} \Delta, \bot}{\Gamma \mathrel{|\!\sim} \Delta}\ (\bot)$$

$$\frac{\Gamma \mathrel{|\!\sim} \Delta, \varphi, \psi}{\Gamma \mathrel{|\!\sim} \Delta, \varphi \vee \psi}\ (\vee_r) \qquad \frac{\Gamma, \varphi \mathrel{|\!\sim} \Delta \qquad \Gamma, \psi \mathrel{|\!\sim} \Delta, \varphi}{\Gamma, \varphi \vee \psi \mathrel{|\!\sim} \Delta}\ (\vee_l) \qquad \frac{\Gamma \mathrel{|\!\sim} \Delta, \varphi}{\Gamma, \neg\varphi \mathrel{|\!\sim} \Delta}\ (\neg_l)$$

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

Figure 2: Basic rules of the sequent calculus

Example 2. *Consider the theory* $(a \vee b) \wedge (b \supset a)$ *which has one equilibrium model* $\{a\}$. *Hence, it holds that* $\neg b$ *is equilibrium entailed.*

$$\dfrac{\dfrac{\dfrac{a \mathrel{|\!\sim} \neg b \qquad a \mathrel{|\!\sim} \neg b, b}{a, b \supset a \mathrel{|\!\sim} \neg b}\ (\supset_l) \qquad \dfrac{b, a \mathrel{|\!\sim} \neg b, a \qquad b \mathrel{|\!\sim} \neg b, b}{b, b \supset a \mathrel{|\!\sim} \neg b, a}\ (\supset_l)}{a \vee b, b \supset a \mathrel{|\!\sim} \neg b}\ (\vee_l)}{(a \vee b) \wedge (b \supset a) \mathrel{|\!\sim} \neg b}\ (\wedge_l)$$

The reason why the implication rule is different from the classical one is that by putting $\neg\varphi$ on the right-hand side of the left premise, we can show that uneven negative cycles lead to instability.

Example 3. *We will show that* $\neg p \supset p \approx \bot$, *i.e., has no equilibrium models.*

$$\dfrac{\dfrac{\dfrac{p \mathrel{|\!\sim} \bot, p}{p, \neg p \mathrel{|\!\sim} \bot}\ (\neg_l)}{p \mathrel{|\!\sim} \bot, \neg\neg p}\ (\neg_r) \qquad \mathrel{|\!\sim} \bot, \neg p}{\neg p \supset p \mathrel{|\!\sim} \bot}\ (\supset_l)$$

The following restricted version of the classical cut rule stems from an axiomatisation of propositional circumscription [Olivetti, 1992].

$$\frac{\Gamma \mathrel{|\!\sim} \Delta, \varphi \qquad \Gamma, \varphi \mathrel{|\!\sim} \Sigma}{\Gamma \mathrel{|\!\sim} \Delta, \Sigma}\ (RCut)$$

As it turns out, $(RCut)$ is also admissible in our case.

Proposition 5. *The (RCut)-rule is sound for equilibrium entailment, i.e., if* $\Gamma \approx \Delta, \varphi$ *and* $\Gamma, \varphi \approx \Sigma$, *then* $\Gamma \approx \Delta, \Sigma$.

Proof. Towards a contradiction, suppose $\Gamma \not\approx \Delta, \Sigma$, then there is some equilibrium model I of Γ such that $I \not\models \Delta$ and $I \not\models \Sigma$. The former and $\Gamma \approx \Delta, \varphi$ imply that $I \models \varphi$. Since $\Gamma, \varphi \approx \Sigma$ and $I \not\models \Sigma$, there has to be some $J \subset I$ such that $\langle J, I \rangle \models \Gamma \cup \{\varphi\}$. However, the latter implies $\langle J, I \rangle \models \Gamma$ contradicting I being an equilibrium model of Γ. □

The rules mentioned so far are not enough to achieve a complete axiomatisation of equilibrium entailment, for example, we cannot show that $a \supset b, b \supset a \approx \neg a$ holds. Intuitively, the issue here is that a is not in any equilibrium model, because it only has cyclic support, but our rules cannot capture this notion. The rule given in Figure 3 fills this gap and makes use of an anti-sequent calculus, cf., Section 2 for details.

Proposition 6. *Given a theory* Γ, $S \subseteq \mathit{Var}(\Gamma)$ *and* $\overline{S} = \mathit{Var}(\Gamma) \setminus S$. *If (a)* $\Gamma \approx \bigwedge S \cup \neg\overline{S}$ *and (b)* $\Gamma, \neg\neg S, \neg\overline{S} \not\models_{HT} \bigwedge_{p \in S} p \vee \neg p$, *then* $\Gamma \approx \Delta$ *for every* Δ.

Proof. Towards a contradiction, suppose there is some equilibrium model I of Γ. Now, from (a) it follows that $I = S$ and from (b) we obtain that there is some $\langle J', I' \rangle \models \Gamma$ where $I' = S = I$ and $\langle J', I' \rangle \not\models p \vee \neg p$ for some $p \in S$. The latter implies $p \notin J'$ and thus $J' \subset S$ which in turn implies that I cannot be an equilibrium model of Γ. □

The intuition is that the left premises ensures that whenever Γ has an equilibrium model that model is S. The refutation in the right-premise gives us that S is a classical model but unstable and Γ has thus no equilibrium models.

Proposition 6 is also closely related to the concept of *safe beliefs* [Osorio *et al.*, 2005], which coincide with equilibrium models and are defined in a similar fashion.

Example 4. *Consider the theory* $\Gamma_1 = \{a \supset b, b \supset a\}$ *which has the empty set as its sole equilibrium model. Clearly,* $\Gamma_1 \approx \neg a$ *holds and Figure 4 shows the corresponding proof.*

Definition 6. *The sequent calculus* ELK *is defined by the initial sequents given in Definition 4, the basic rules in Figure 2, the* $(RCut)$*-rule, and the* $(\dashv)$*-rule.*

The introduced rules constitute our sequent calculus ELK.

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

Having established soundness, it remains to show that the calculus is complete. First, introduce the notion of consistency as well as some properties of the calculus.

Definition 7. Γ *is* equilibrium inconsistent *whenever* $\Gamma \mathrel{|\!\sim} \bot$ *can be derived, in* ELK *and* equilibrium consistent *otherwise.*

We drop "equilibrium" here whenever clear from context.

Proposition 7. $\Gamma \mathrel{|\!\sim} \varphi$ *can be derived iff* $\Gamma \cup \{\neg\varphi\}$ *is inconsistent.*

Proof. Suppose $\Gamma \mathrel{|\!\sim} \varphi$ can be derived, by application of the rule $(\neg_l)$ we can derive $\Gamma, \neg\varphi \mathrel{|\!\sim}$ and thus $\Gamma, \neg\varphi \mathrel{|\!\sim} \bot$ using (w_r). With $\Gamma, \neg\varphi \mathrel{|\!\sim} \bot$ we can derive $\Gamma \mathrel{|\!\sim} \varphi$ as follows:

$$\dfrac{\dfrac{\Gamma, \varphi \mathrel{|\!\sim} \varphi}{\Gamma \mathrel{|\!\sim} \varphi, \neg\varphi}\ (\neg_r) \qquad \dfrac{\Gamma, \neg\varphi \mathrel{|\!\sim} \bot}{\Gamma, \neg\varphi \mathrel{|\!\sim}}\ (\bot)}{\Gamma \mathrel{|\!\sim} \varphi}\ (RCut)$$

$$\frac{\Gamma \mathrel{|\!\sim} \bigwedge S \cup \neg\overline{S} \qquad \Gamma, \neg\neg S, \neg\overline{S} \dashv_{HT} \bigwedge_{p\in S} p \vee \neg p}{\Gamma \mathrel{|\!\sim} \Delta}\ (\dashv) \text{ where } S \subseteq Var(\Gamma) \text{ and } \overline{S} = Var(\Gamma) \setminus S$$

Figure 3: Refutation-based rule in the sequent calculus

$$\frac{\dfrac{\vdots}{a \supset b, b \supset a \mathrel{|\!\sim} \neg a, \neg\neg a \wedge \neg\neg b} \qquad \dfrac{\dfrac{\vdots}{a \supset b, b \supset a, \neg\neg a, \neg\neg b \mathrel{|\!\sim} a \wedge b} \qquad \dfrac{\vdots}{a \supset b, b \supset a, \neg\neg a, \neg\neg b \dashv_{HT} (a \vee \neg a) \wedge (b \vee \neg b)}}{a \supset b, b \supset a, \neg\neg a, \neg\neg b \mathrel{|\!\sim} \neg a}\ (\dashv)}{a \supset b, b \supset a \mathrel{|\!\sim} \neg a}\ (RCut)$$

Figure 4: Partial derivation of $a \supset b, b \supset a \mathrel{|\!\sim} \neg a$

□

Proposition 8. *If $\Gamma \mathrel{|\!\sim} \varphi$ can be derived and $\neg\varphi \in \Gamma$, then Γ is inconsistent.*

Proof. Suppose $\Gamma \mathrel{|\!\sim} \varphi$ can be derived and $\neg\varphi \in \Gamma$, then by application of the rules $(\neg_l)$ and (w_r) we can derive $\Gamma, \neg\varphi \mathrel{|\!\sim} \bot$ and since by definition the sides of equilibrium sequents are sets, we obtain $\Gamma \mathrel{|\!\sim} \bot$. □

Proposition 9. *If $\varphi \in \Gamma$ and $\neg\varphi \in \Gamma$, then Γ is inconsistent.*

Proof. If $\varphi \in \Gamma$, then $\Gamma \mathrel{|\!\sim} \varphi$ is an initial sequent and Γ is thus inconsistent by Proposition 8. □

Proposition 10. *If $\Gamma \cup \{\varphi\}$ and $\Gamma \cup \{\neg\varphi\}$ are inconsistent, then Γ is inconsistent as well.*

Proof. If $\Gamma, \varphi \mathrel{|\!\sim} \bot$ and $\Gamma, \neg\varphi \mathrel{|\!\sim} \bot$ can be derived, we can derive $\Gamma \mathrel{|\!\sim} \bot$ as follows:

$$\frac{\dfrac{\Gamma, \varphi \mathrel{|\!\sim} \bot}{\Gamma \mathrel{|\!\sim} \bot, \neg\varphi}\ (\neg_r) \qquad \Gamma, \neg\varphi \mathrel{|\!\sim} \bot}{\Gamma \mathrel{|\!\sim} \bot}\ (RCut)$$

□

Our completeness proof will follow the well-known Lindenbaum approach also used in classical logic. Hence, we must define what we understand by a *complete* theory.

Definition 8. *A theory Γ is* complete *iff for every formula φ, either $\neg\neg\varphi \in \Gamma$ or $\neg\varphi \in \Gamma$.*

Different from classical logic, we do not require that each formula occurs either positively or negatively and rather that the negation or the double negation is contained. This is due to the fact that equilibrium logic is not *cumulative*, i.e., adding positive formulas can induce new equilibrium models. However, as the next results show, this is not the case for negated or double negated formulas.

Lemma 1. *If I is an equilibrium model of Γ and $I \models \varphi$, then I is an equilibrium model of $\Gamma \cup \{\varphi\}$.*

Proof. Suppose I is not an equilibrium model of $\Gamma \cup \{\varphi\}$. It is is clearly a classical model, as $I \models \Gamma$ follows from it being an equilibrium model of Γ and $I \models \varphi$. Hence, there is some $J \subset I$ such that $\langle J, I\rangle \models \Gamma \cup \{\varphi\}$. The latter implies $\langle J, I\rangle \models \Gamma$, so I is not an equilibrium model of Γ. □

Lemma 2. *If $I \models \neg\varphi$, then I is an equilibrium model of Γ iff it is an equilibrium model of $\Gamma \cup \{\neg\varphi\}$.*

Proof. The left-to-right direction, follows from Lemma 1 and $I \models \neg\varphi$. Remains to show the other direction. Towards a contradiction, suppose I is not an equilibrium model of Γ. Since it is a classical model, there has to be some $J \subset I$ such that $\langle J, I\rangle \models \Gamma$. However, $I \models \neg\varphi$ implies $\langle J, I\rangle \models \neg\varphi$. Hence, $\langle J, I\rangle \models \Gamma \cup \{\neg\varphi\}$ holds contradicting that I is an equilibrium model of $\Gamma \cup \{\neg\varphi\}$. □

Lemma 3. *If $I \models \varphi$, then I is an equilibrium model of Γ iff it is an equilibrium model of $\Gamma \cup \{\neg\neg\varphi\}$.*

Proof. Follows from Lemma 2. □

Establishing that every complete theory has some classical model works similarly to the classical case and leads to the following lemma.

Lemma 4. *Given a complete theory Γ, if Γ is consistent, then there is some interpretation I such that $I \models \Gamma$.*

Furthermore, the classical model is unique.

Lemma 5. *Given a complete theory Γ. Whenever $\langle J, I\rangle \models \Gamma$ and $\langle J', I'\rangle \models \Gamma$, then $I = I'$.*

Proof. Suppose $\langle J, I\rangle \models \Gamma$ and $\langle J', I'\rangle \models \Gamma$. We show $I \subseteq I'$. Let p be an arbitrary element of I. Since Γ is complete, either $\neg\neg p \in \Gamma$ or $\neg p \in \Gamma$. Now, if $\neg p \in \Gamma$, then $\langle J, I\rangle \models \neg p$ holds, which implies $I \models \neg p$ and thus $p \notin I$. So, suppose $\neg\neg p \in \Gamma$ and thus $\langle J', I'\rangle \models \neg\neg p$ which implies $I' \models \neg\neg p$. Hence, $p \in I'$. The statement $I' \subseteq I$ can be shown mutatis mutandis and thus $I = I'$. □

Of course, having a classical model is not enough in our case. We also need the following lemma, which shows that our calculus enforces stability.

Lemma 6. *Given a complete theory Γ. If $\langle J, I\rangle \models \Gamma$ where $J \subset I$, then Γ is inconsistent.*

Proof. Let $S = I$ and $\overline{S} = \mathit{Var}(\Gamma) \setminus S$. Since Γ is complete, $\neg p \in \Gamma$ for every $p \in \overline{S}$ and we can thus derive $\Gamma \mathrel{|\!\sim} \neg p$ as it is an initial sequent. Similarly, $\neg\neg p \in \Gamma$ for every $p \in S$ and we can derive $\Gamma \mathrel{|\!\sim} p$ by applications of the $(\neg_l)$ and $(\neg_r)$ rules. Hence, $\Gamma \mathrel{|\!\sim} \bigwedge S \cup \neg\overline{S}$ can be derived through $(\wedge_r)$. Furthermore, $S = I$, $\langle J, I\rangle \models \Gamma$, $p \in I \setminus J$ and $p \in S$ imply $\Gamma, \neg\neg S, \neg\overline{S} \not\models_{HT} \bigwedge_{p \in S} p \vee \neg p$. Hence, $\Gamma \mathrel{|\!\sim} \bot$ follows from the $(\dashv)$-rule and the completeness of the HT anti-sequent calculus. □

Finally, we can state the following which follows immediately from the lemmas above.

Corollary 1. *Given a complete theory* Γ*, if* Γ *is consistent, then it has an equilibrium model.*

Now, having the tools in place, we come to the actual Lindenbaum-like construction. First, another helpful lemma.

Lemma 7. *Given a consistent theory* Γ*, if* $\Gamma \mathrel{|\!\sim} \varphi$ *can be derived, then* $\Gamma \cup \{\varphi\}$ *is consistent.*

Proof. Suppose $\Gamma \mathrel{|\!\sim} \varphi$ can be derived, then by Proposition 7, $\Gamma \cup \{\neg\varphi\}$ is inconsistent. Towards a contradiction, assume $\Gamma \cup \{\varphi\}$ is inconsistent. Then by Proposition 10, Γ is inconsistent. □

Proposition 11. *Given a consistent theory* Γ *there exists* $\Gamma^* \supseteq \Gamma$ *which is consistent and complete.*

Proof. Let $\varphi_0, \ldots, \varphi_n$ be an enumeration of all formulas in the language. We define $\Gamma_0 = \Gamma$ and

$$\Gamma_{i+1} = \begin{cases} \Gamma_i \cup \{\neg\neg\varphi_i\} & \text{if } \Gamma_i \cup \{\neg\neg\varphi_i\} \text{ is consistent, and} \\ \Gamma_i \cup \{\neg\varphi_i\} & \text{otherwise.} \end{cases}$$

Γ_0 is consistent by assumption. Suppose $\Gamma_{i+1} = \Gamma_i \cup \{\neg\neg\varphi_i\}$, then Γ_{i+1} is consistent by construction. Now, if $\Gamma_{i+1} = \Gamma_i \cup \{\neg\varphi_i\}$ then $\Gamma_i \cup \{\neg\neg\varphi_i\}$ is inconsistent and therefore, by Proposition 7, $\Gamma \mathrel{|\!\sim} \neg\varphi$ can be derived. By Lemma 7, the latter and Γ_i being consistent imply $\Gamma_i \cup \{\neg\varphi_i\} = \Gamma_{i+1}$ is consistent. Hence, $\Gamma^* = \bigcup_{1 \le i \le n} \Gamma_i$ is consistent.

By construction, Γ^* is complete as every formula φ_i appears in the enumeration and either $\neg\neg\varphi_i$ of $\neg\varphi_i$ gets added. □

The completeness of the calculus now follows easily.

Theorem 2. *If* $\Gamma \approx \varphi$*, then* $\Gamma \mathrel{|\!\sim} \varphi$ *is derivable in* ELK.

Proof. We proceed by contraposition. Suppose $\Gamma \mathrel{|\!\sim} \varphi$ cannot be derived. By Proposition 7 it holds that $\Gamma \cup \{\neg\varphi\}$ is consistent. Hence, it has some complete extension which has an equilibrium model I such that $I \models \neg\varphi$. By Lemmas 2 and 3, I is also an equilibrium model of Γ and $I \models \neg\varphi$ clearly implies $I \not\models \varphi$. Hence, $\Gamma \not\approx \varphi$ holds by definition. □

The sequent calculus ELK thus satisfies the following.

Corollary 2 (Main Result)**.** *The sequent calculus* ELK *is sound and complete.*

5 Discussion

Alternative Characterisation. Naturally, one might wonder whether there is a way to replace the $(\dashv)$-rule and ideally with a rule that does not rely on a separate refutation calculus.

The entailment in Example 4 is shown in the MLK sequent calculus for circumscription [Olivetti, 1992] using the *weak monotonicity* rule; the encoded property of inference is also known as *cautious monotonicity (CM)*. It is well-known that answer set entailment does not satisfy it in general, but in a restricted syntactic fragment.

In the following, we call a theory *nested* if it contains nested implication, i.e., an implication $\varphi \supset \psi$ where either φ or ψ contains an implication as a subformula, and negation $\neg\chi$ is interpreted as $\chi \supset \bot$ and thus counts as an implication as well.

With this in place, we can introduce the following rule.

$$\frac{\Gamma \mathrel{|\!\sim} \Delta \qquad \Gamma \mathrel{|\!\sim} \varphi}{\Gamma, \varphi \mathrel{|\!\sim} \Delta}\ (RCM) \text{ if } \Gamma \text{ and } \varphi \text{ are not nested}$$

This rule is sound for equilibrium entailment.

Proposition 12. *Given a theory* Γ *that is not nested, if* $\Gamma \approx \Delta$ *and* $\Gamma \approx \varphi$ *hold, then so does* $\Gamma, \varphi \approx \Delta$.

The proof follows from the fact that for non-nested theories, minimal models and equilibrium models coincide.

Furthermore, consider the following rule, where $\vdash_{HT}$ denotes an HT sequent calculus, e.g. the one given by Mints [2010], and let $\leftrightarrow$ denote biimplication.

$$\frac{\Gamma, \psi \mathrel{|\!\sim} \Delta \qquad \vdash_{HT} \varphi \leftrightarrow \psi}{\Gamma, \varphi \mathrel{|\!\sim} \Delta}\ (LLE)$$

Intuitively, the rule states that HT-equivalent subformulas are interchangeable, which is a well-known property of equilibrium logic [Lifschitz *et al.*, 2001].

Using the two rules, we obtain the following result.

Theorem 3. *Let* ELK′ *denote the calculus of initial sequents from Definition 4, the rules in Figure 2,* $(RCut)$*,* (RCM)*, and* (LLE)*. Then,* $\Gamma \approx \varphi$ *iff* $\Gamma \mathrel{|\!\sim} \varphi$ *is derivable in* ELK′.

The basic idea is that for each complete theory, there is an HT-equivalent theory which consists of a non-nested part and negated or double-negated atoms.

Hence, (RCM) and (LLE) replace the $(\dashv)$-rule, but has rather drastic syntactic side conditions and we have to utilize an HT sequent calculus.

ASP Fragment. Given that part of our motivation for this work is to provide a proof system for ASP, it is natural to ask how our calculus behaves on this fragment of EL. Recall that programs are theories where every formula is an implication $l_1 \wedge \cdots \wedge l_m \supset l_{m+1} \vee \cdots \vee l_n$ and all l_i are literals.

We have already seen in Example 4 that even for programs, the $(\dashv)$ rule may be needed to derive certain consequences. However, as we show there is an important class of programs for which this is not the case. In fact, not even the $(RCut)$ rule is needed for it. The class in question is the one of *tight* programs which are defined as follows. A program is *tight* [Erdem and Lifschitz, 2003] if there is no loop in its *dependency graph*, which is constructed over all atoms by adding an edge from p to q for each positive literals p and q in the antecedent and the consequent of a rule, respectively.[1]

The intuition is that for tight programs, the $(\supset_l)$-rule becomes invertible, as the next result shows, which is not the case in general.

Proposition 13. *If $\Gamma, \varphi \supset \psi \approx \Delta$ is satisfied, where Γ is a tight program, then $\Gamma, \psi \approx \Delta, \neg\varphi$ and $\Gamma \approx \Delta, \varphi$.*

Using this proposition, the following can be shown.

Theorem 4. *Given tight ASP program Γ and Δ in negation normal form, then $\Gamma \approx \Delta$ holds iff $\Gamma \sim \Delta$ can be derived using the basic rules in Figure 2, and initial sequents from Definition 4.*

Proof (Sketch). It can be seen that for programs only the $(\wedge_l)$, $(\vee_l)$, $(\neg_l)$ and $(\supset_l)$ rules are ever required for the left-hand side of the sequent. Furthermore, for tight programs, the $(\supset_l)$ rule is invertible by Proposition 13. The rules $(\wedge_l)$ and $(\neg_l)$ are in invertible in general already. On the left-hand sides we now only have sets of clauses and the selection of the branches for the $(\vee)_l$ rule is guided by which literal satisfies the most clauses. Furthermore, by deriving the rules only after their bodies have been added to the left, also the sometimes required $(\neg_r)$-rule becomes invertible.

We thus always have leaves $\Gamma' \sim \Delta'$ where Γ' is a set of atoms and Δ' is a set of literals and if $\Gamma \approx \Delta$ holds then the sequents in the leaves have to be axioms due to invertibility. □

Given that for tight programs, answer sets and so-called *supported models* coincide [Erdem and Lifschitz, 2003], Theorem 4 implies that the rules given in Figure 2 axiomatise the inference over such models. This further suggests that in the full calculus, the $(\dashv)$ rule is required to exclude supported models which include *unfounded sets* [Leone *et al.*, 1997], i.e., atoms which are deemed true but only have cyclic support.

The observation above also gives rise to a general proof strategy when dealing with programs. First, one should apply the $(\supset_l)$-rule on parts of the program which are *locally tight*, i.e., the dependency graph of that program module is non-cyclic, leaving the $(\dashv)$-rule to potentially handle remaining loops. Due to space restrictions we cannot go into more details, but proof strategies also arise by utilisation of *splitting sets* [Lifschitz and Turner, 1994].

Explanations. We argued in Section 3 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 will discuss, they serve as an overarching theoretical framework from which explanations can be derived. To that end we can also introduce derived rules, which may be less general but help shorten proofs. For example, we use the following rule, which essentially encodes *modus ponens* and we will see later that this rule is very useful when one considers ASP programs.

$$\frac{\Gamma, \varphi, \psi \sim \Delta}{\Gamma, \varphi, \varphi \supset \psi \sim \Delta}\ (MP)$$

In Example 3, we have already seen how the calculus shows why a theory is inconsistent. In that proof, we see that since $\neg p$ holds by default and p entails p, $\neg p \supset p$ cannot have any equilibrium models.

Let us consider some more examples with theories that essentially represent ASP programs.

Example 5. *Consider $\Gamma_1 = \{\neg c \supset (a \vee b), \neg(a \wedge d), \neg b, d\}$. The atom d can be seen as the input data and the remaining formulas encode a choice between a and b given $\neg c$ and some constraints. It holds that $\Gamma_1 \approx \bot$, i.e., has no equilibrium models which we can prove as shown in Figure 5.*

Let us verbalise the steps of the proof bottom up. First, we select $\neg c \supset (a \vee b)$ and detach the implication with the $(\supset_l)$-rule. In the right branch, we now need to show that $\neg c$ follows. After two applications of $(\neg_l)$ we only have atoms on the left and have an axiom. On the left branch, we have added $a \vee b$ on the left and proceed by case distinction through the $(\vee_l)$-rule.

In the branch where a is now on the left, we also have d. We can apply $(\neg_l)$ to $\neg(a \wedge d)$ and get $a \wedge d$ on the right. The latter can be split with the $(\wedge_r)$-rule and since both a and d were already on the left, we have two axioms. Otherwise, if b holds, then this clashes with $\neg b$ which is shown the same as for the other branch.

As we have seen in the example, a proof in the sequent calculus which is summarised and, put into natural language, can serve as an explanation.

Let us now turn to an example where we have a theory and one of its equilibrium models.

Example 6. *Let $\Gamma_2 = \{\neg a \supset b, \neg b \supset a, a \supset c\}$ and consider its equilibrium model $I = \{a, c\}$. Suppose we seek to explain why c is in the model. A minimal assumption set w.r.t. I, cf. Section 3, is $\{b\}$ and we give the following sequent proof:*

$$\dfrac{\dfrac{\neg a \supset b, \neg b, a, c \sim c}{\neg a \supset b, a \supset c, \neg b, a \sim c}\ (MP)}{\neg a \supset b, \neg b \supset a, a \supset c, \neg b \sim c}\ (MP)$$

The proof can be summarised as follows. We look at the implication $\neg b \supset a$. Since $\neg b$ is in the theory, we can apply the modus ponens rule (MP) from above. With a in the theory, we can again use (MP) on $a \supset c$. Given that c is now on the left and right, we have an axiom.

So far we have only used the rules from ELK and (MP). However, one might introduce further derived rules and use them in explanations. A rule akin to Proposition 2 comes to mind. Furthermore, in a potential explanation system, lemmas might be reused using the $(RCut)$-rule, shortening the proofs.

6 Related Work

To the best of our knowledge, no sequent calculus for equilibrium logic has been given in the literature. Pearce *et al.* [2000] introduced a tableau system for equilibrium logic. Their approach works in several stages. In the first stage, a dedicated

[1] Tight programs are usually considered to have no negative literals in rule heads. However, relevant concepts, like supporting rules, carry over to the more general case.

$$
\frac{\dfrac{\dfrac{\dfrac{a,\neg b,d \mathrel{|\!\sim} \bot,\neg\neg c,a \qquad a,\neg b,d \mathrel{|\!\sim} \bot,\neg\neg c,d}{a,\neg b,d \mathrel{|\!\sim} \bot,\neg\neg c,a\wedge d}\,(\wedge_r)}{a,\neg(a\wedge d),\neg b,d \mathrel{|\!\sim} \bot,\neg\neg c}\,(\neg_l) \qquad \dfrac{b,\neg(a\wedge d),d \mathrel{|\!\sim} \bot,\neg\neg c,a,b}{b,\neg(a\wedge d),\neg b,d \mathrel{|\!\sim} \bot,\neg\neg c}\,(\neg_l)}{a\vee b,\neg(a\wedge d),\neg b,d \mathrel{|\!\sim} \bot,\neg\neg c}\,(\vee_l) \qquad \dfrac{\dfrac{d \mathrel{|\!\sim} \bot,\neg c,a\wedge d,b}{\neg b,d \mathrel{|\!\sim} \bot,\neg c,a\wedge d}\,(\neg_l)}{\neg(a\wedge d),\neg b,d \mathrel{|\!\sim} \bot,\neg c}\,(\neg_l)}{\neg c\supset(a\vee b),\neg(a\wedge d),\neg b,d \mathrel{|\!\sim} \bot}\,(\supset_l)
$$

Figure 5: Derivation for Example 5

calculus for total HT models is used to enumerate all models. With another dedicated calculus, those total models are then checked for stability in the second stage. Finally, in the last stage, it is checked via tableaux whether the given formula holds in the previously obtained total and stable models. Hence, due to the required enumeration and multi-stage concept, this calculus does not fully satisfy requirements (R1) and (R3) introduced in Section 1.

Several proof systems have been studied directly for ASP. Bonatti [2001] introduced a resolution calculus for answer set entailment and provided another calculus for *brave entailment* [Bonatti *et al.*, 2008]. The latter is different from the notion of entailment we study in the sense that the consequence does not need to hold in all models but rather in at least one. The calculi differ from their classical counterparts and operate on pairs of sets of literals and suffer from some drawbacks: they support only normal logic programs and rely on a rather complicated notion of *counter-supports* which is not axiomatised by the calculus but computed externally. The calculi are thus weak on requirements (R1-R3).

Gebser and Schaub [2013] presented a tableau calculus which is aimed at axiomatising satisfiability and how solvers obtain their solutions and thus differs in motivation from our work. However, it can also be used for answer set entailment by translation of the entailment into an inconsistent program. While their calculus supports a large number of advanced ASP language features like disjunction and weight constraints, it does not cover arbitrary nested formulas like our calculus does. Furthermore, some rules in their calculus heavily rely on global syntactic notions which only work for programs, which clashes with our requirement (R1).

Proofs for ASP also appear in the context of proof logging that ASP solvers provide to justify whenever they report unsatisfiability [Alviano *et al.*, 2019; Chew *et al.*, 2024]. However, those proofs are generally not geared towards human interpretability but rather verification and are not very concise. Thus, they are violating requirement (R3).

Proof-like systems have been used to provide explanations as to why certain atoms are, or are not, in a given answer set [Pontelli *et al.*, 2009; Alviano *et al.*, 2024]. While the explanations given by those approaches share some similarity with a formal proof, they are not actual complete proof systems and cannot be applied without a model. Furthermore, they are limited to the basic ASP language.

Similarly, the fix-point characterisation for equilibrium logic [Pearce, 2006] does not axiomatise the inference relation over equilibrium models, but rather only provides a definition of stable models via HT entailment and model selection. The characterisation can be used in conjunction with an HT sequent calculus [Mints, 2010] to justify the atoms in an equilibrium model in a proof-theoretic manner. However, this is not the scope of our work.

Finally, we mention some inspiring related sequent calculi that axiomatise other nonmonotonic formalism, viz. the one by Olivetti [1992] for propositional circumscription, which is also two-sided, and those by Bonatti and Olivetti for Circumscription [Bonatti and Olivetti, 1997b] and Default Logic [Bonatti and Olivetti, 1997a], which also use an anti-sequent calculus.

7 Conclusion

In this work, 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 to ensure the stability condition necessary for equilibrium models.

We also discussed an alternative to this refutation rule which is based on cautious monotonicity. However, this property only holds for a syntactic fragment of the logic that includes positive ASP programs. However, it is unlikely that there is an axiomatisation which operates on purely standard rules. We further showed how answer set entailment can serve as a framework for explainability and how our calculus can in turn justify why an entailment holds.

For future work, we plan to further explore the use of proof-based explanations for ASP, for example, by studying translations of proofs into natural language including proof search and human-friendly presentation of proofs, or investigating contrastive explanations. Extensions of the calculus to cover more general versions of equilibrium logic, which encompass programs with variables [Pearce and Valverde, 2008] or linear constraints [Cabalar *et al.*, 2016] are also of interest.

Acknowledgments

Tobias Geibinger is a recipient of a DOC Fellowship of the Austrian Academy of Sciences at the Institute of Logic and Computation at the TU Wien.

This work has benefited from the European Union's Horizon 2020 research and innovation programme under grant agreement No 101034440 (LogiCS@TUWien).

In addition, this research was funded in part by the Austrian Science Fund (FWF) [10.55776/COE12].

Furthermore, we thank Agata Ciabattoni, Matteo Tesi, David Pearce and Hans Tompits for the helpful discussions on the topic.

References

[Abels *et al.*, 2019] Dirk Abels, Julian Jordi, Max Ostrowski, Torsten Schaub, Ambra Toletti, and Philipp Wanko. Train Scheduling with Hybrid ASP. In *Proceedings of the 15th International Conference on Logic Programming and Nonmonotonic Reasoning (LPNMR 2019)*, volume 11481 of *LNCS*, pages 3–17. Springer, 2019.

[Ali *et al.*, 2023] Ramsha Ali, Mohammed M. S. El-Kholany, and Martin Gebser. Flexible job-shop scheduling for semiconductor manufacturing with hybrid answer set programming (application paper). In *Proceedings of the 25th International Symposium on Practical Aspects of Declarative Languages (PADL 2023)*, pages 85–95, Cham, 2023. Springer.

[Alviano *et al.*, 2013] Mario Alviano, Carmine Dodaro, Wolfgang Faber, Nicola Leone, and Francesco Ricca. 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*, pages 54–66. Springer, 2013.

[Alviano *et al.*, 2019] Mario Alviano, Carmine Dodaro, Johannes Klaus Fichte, Markus Hecher, Tobias Philipp, and Jakob Rath. Inconsistency proofs for ASP: the ASP - DRUPE format. *Theory and Practice of Logic Programming*, 19(5-6):891–907, 2019.

[Alviano *et al.*, 2024] Mario Alviano, Susana Hahn, Orkunt Sabuncu, and Hannes Weichelt. Answer set explanations via preferred unit-provable unsatisfiable subsets. In *Proceedings of the 17th International Conference on Logic Programming and Nonmonotonic Reasoning (LPNMR 2024)*, volume 15245 of *LNCS*, pages 187–199. Springer, 2024.

[Beierle *et al.*, 2005] Christoph Beierle, Oliver Dusso, and Gabriele Kern-Isberner. Using answer set programming for a decision support system. In *Proceedings of the 8th International Conference (LPNMR 2005)*, volume 3662 of *LNCS*, pages 374–378. Springer, 2005.

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

[Bonatti and Olivetti, 1997b] Piero A. Bonatti and Nicola Olivetti. A sequent calculus for skeptical default logic. In *Proceedings of the 6th International Conference on Automated Reasoning with Analytic Tableaux and Related Methods (TABLEAUX 1997)*, volume 1227 of *LNCS*, pages 107–121. Springer, 1997.

[Bonatti *et al.*, 2008] Piero A. Bonatti, Enrico Pontelli, and Tran Cao Son. Credulous resolution for answer set programming. In *Proceedings of the 23rd AAAI Conference on Artificial Intelligence (AAAI 2008)*, pages 418–423. AAAI Press, 2008.

[Bonatti, 2001] Piero A. Bonatti. Resolution for skeptical stable model semantics. *Journal of Automated Reasoning*, 27(4):391–421, 2001.

[Cabalar *et al.*, 2016] Pedro Cabalar, Roland Kaminski, Max Ostrowski, and Torsten Schaub. An ASP semantics for default reasoning with constraints. In *Proceedings of the 25th International Joint Conference on Artificial Intelligence (IJCAI 2016)*, pages 1015–1021. IJCAI/AAAI Press, 2016.

[Chew *et al.*, 2024] Leroy Chew, Alexis de Colnet, and Stefan Szeider. ASP-QRAT: A conditionally optimal dual proof system for ASP. In *Proceedings of the 21st International Conference on Principles of Knowledge Representation and Reasoning (KR 2024)*, 2024.

[Comploi-Taupe *et al.*, 2022] Richard Comploi-Taupe, Giulia Francescutto, and Gottfried Schenner. Applying incremental answer set solving to product configuration. In *Proceedings of the 26th ACM International Systems and Software Product Line Conference (SPLC 2022)*, pages 150–155. ACM, 2022.

[Eiter *et al.*, 1997] Thomas Eiter, Georg Gottlob, and Nicola Leone. Abduction from logic programs: Semantics and complexity. *Theoretical Computer Science*, 189(1-2):129–177, 1997.

[Eiter *et al.*, 1999] Thomas Eiter, Wolfgang Faber, Nicola Leone, and Gerald Pfeifer. The diagnosis frontend of the dlv system. *AI Communications*, 12(1-2):99–111, 1999.

[Erdem and Lifschitz, 2003] Esra Erdem and Vladimir Lifschitz. Tight logic programs. *Theory and Practice of Logic Programming*, 3(4-5):499–518, 2003.

[Erdem and Oztok, 2015] Esra Erdem and Umut Oztok. Generating explanations for biomedical queries. *Theory and Practice of Logic Programming*, 15(1):35–78, 2015.

[Erdem *et al.*, 2016] Esra Erdem, Michael Gelfond, and Nicola Leone. Applications of answer set programming. *AI Magazine*, 37(3):53–68, 2016.

[Falkner *et al.*, 2018] Andreas Falkner, Gerhard Friedrich, Konstantin Schekotihin, Richard Taupe, and Erich C. Teppan. Industrial applications of answer set programming. *KI - Künstliche Intelligenz*, 32(2):165–176, 2018.

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

[Gebser and Schaub, 2013] Martin Gebser and Torsten Schaub. Tableau calculi for logic programs under answer set semantics. *ACM Transactions on Computational Logic*, 14(2):15:1–15:40, 2013.

[Gebser *et al.*, 2019] Martin Gebser, Roland Kaminski, Benjamin Kaufmann, and Torsten Schaub. Multi-shot ASP solving with clingo. *Theory and Practice of Logic Programming*, 19(1):27–82, 2019.

[Gentzen, 1935] Gerhard Gentzen. Untersuchungen über das logische Schließen. I. *Mathematische Zeitschrift*, 39:176–210, 1935.

[Inclezan, 2015] Daniela Inclezan. An application of answer set programming to the field of second language acquisition. *Theory and Practice of Logic Programming*, 15(1):1–17, 2015.

[Leone *et al.*, 1997] Nicola Leone, Pasquale Rullo, and Francesco Scarcello. Disjunctive Stable Models: Unfounded Sets, Fixpoint Semantics and Computation. *Information and Computation*, 135(2):69–112, June 1997.

[Leone *et al.*, 2006] Nicola Leone, Gerald Pfeifer, Wolfgang Faber, Thomas Eiter, Georg Gottlob, Simona Perri, and Francesco Scarcello. The DLV system for knowledge representation and reasoning. *ACM Transactions on Computational Logic*, 7(3):499–562, 2006.

[Lifschitz and Turner, 1994] Vladimir Lifschitz and Hudson Turner. Splitting a logic program. In *Proceedings of the 11th International Conference on Logic Programming (ICLP 1994)*, pages 23–37. MIT Press, 1994.

[Lifschitz *et al.*, 2001] Vladimir Lifschitz, David Pearce, and Agustín Valverde. Strongly equivalent logic programs. *ACM Transactions on Computational Logic*, 2(4):526–541, 2001.

[Mints, 2010] Grigori Mints. Cut-free formulations for a quantified logic of here and there. *Annals of Pure and Applied Logic*, 162(3):237–242, 2010.

[Oetsch and Tompits, 2011] Johannes Oetsch and Hans Tompits. Gentzen-type refutation systems for three-valued logics with an application to disproving strong equivalence. In *Proceedings of the 11th International Conference on Logic Programming and Nonmonotonic Reasoning (LPNMR 2011)*, volume 6645 of *LNCS*, pages 254–259. Springer, 2011.

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

[Osorio *et al.*, 2005] Mauricio Osorio, Juan Antonio Navarro Pérez, and José Arrazola. Safe beliefs for propositional theories. *Annals of Pure and Applied Logic*, 134(1):63–82, 2005.

[Pearce and Valverde, 2008] David Pearce and Agustín Valverde. Quantified equilibrium logic and foundations for answer set programs. In *Proceedings of the 24th International Conference on Logic Programming (ICLP 2008)*, volume 5366 of *LNCS*, pages 546–560. Springer, 2008.

[Pearce *et al.*, 2000] David Pearce, Inman P. de Guzmán, and Agustín Valverde. A tableau calculus for equilibrium entailment. In *Proceedings of the 9th International Conference on Automated Reasoning with Analytic Tableaux and Related Methods (TABLEAUX 2000)*, volume 1847 of *LNCS*, pages 352–367. Springer, 2000.

[Pearce, 1996] David Pearce. A new logical characterisation of stable models and answer sets. In *Selected Papers of the 2nd International Workshop on Non-Monotonic Extensions of Logic Programming (NMELP 1996)*, volume 1216 of *LNCS*, pages 57–70. Springer, 1996.

[Pearce, 2006] David Pearce. Equilibrium logic. *Annals of Mathematics and Artificial Intelligence*, 47:3–41, 2006.

[Pontelli *et al.*, 2009] Enrico Pontelli, Tran Cao Son, and Omar El-Khatib. Justifications for logic programs under answer set semantics. *Theory and Practice of Logic Programming*, 9(1):1–56, 2009.

[Yli-Jyrä *et al.*, 2023] Anssi Yli-Jyrä, Masood Feyzbakhsh Rankooh, and Tomi Janhunen. Pruning redundancy in answer set optimization applied to preventive maintenance scheduling. In *Proceedings of the 25th International Symposium on Practical Aspects of Declarative Languages (PADL 2023)*, pages 279–294, Cham, 2023. Springer.