# Abduction in equilibrium logic

**David Pearce**
DFKI, Saarbrücken
Germany

**Agustín Valverde**
Dept. of Applied Mathematics
University of Málaga, Spain

**Abstract**

Equilibrium logic is a system of nonmonotonic reasoning that generalises answer set semantics for logic programs to a syntactically broader class of theories. The paper discusses the problem of abduction for equilibrium logic, making use of tableaux systems for equilibrium entailment presented in (Pearce, de Guzmán, & Valverde 2000b). An algorithm for generating potential abductive explanations is presented in outline. This work can be considered as extending current frameworks of abductive logic programming in the answer set tradition and at the same time as extending previous work on abduction using semantic tableaux to the situation of a nonmonotonic underlying logic.

## Introduction

This paper is about adding abductive reasoning to a system of nonmonotonic logic that extends logic programming under answer set semantics. The importance of abductive inference in AI problem solving, eg. in diagnosis and planning, is well known: (Kakas & Michael 1998); also abductive methods have long been combined with logic programming, in *abductive logic programming*. The system in which we work is called equilibrium logic and was developed in (Pearce 1997). Currently, implementations of answer set programming are defined for normal logic programs (eg. `smodels`) or disjunctive logic programs (eg. `dlv`) and therefore do not yet embrace the more general syntax over which answer set semantics has been defined, for instance programs with nested expressions in the sense of (Lifschitz, Tang, & Turner 1999). The system `dlv`[1] provides a front-end for abductive diagnostic reasoning. Also abductive logic programming within the context of answer sets has been studied largely (Kakas, Kowalski, & Toni 1993) in the setting of normal and disjunctive programs. Equilibrium logic on the other hand is defined over a general propositional language and is therefore also suitable for dealing with ground theories in a function-free, first-order logic. As shown in (Lifschitz, Pearce, & Valverde 2000) it captures answer set reasoning with general programs in the style of (Lifschitz, Tang, & Turner 1999). The language of equilibrium logic includes both negation-by-default and strong or explicit negation.

Proof systems for equilibrium logic are discussed in (Pearce, de Guzmán, & Valverde 2000b; 2000a). Here we focus on (Pearce, de Guzmán, & Valverde 2000b) in which tableau systems for equilibrium entailment are developed. These are highly appropriate for studying abductive inference, since models of a theory are explicitly represented by branches in a tableau, and the method of abducing formulas in order to explain a given formula from a theory can be regarded as the process of closing open branches in the appropriate tableau.

The idea of using semantic tableaux to capture abductive reasoning is not new. It has been developed for classical logic, for instance, by (Cialdea Mayer & Pirri 1993; Palopoli, Pirri, & Pizzuti 1999; Aliseda 1998). The topic of abductive inference within nonmonotonic logics is less well-developed and to our knowledge tableau methods have not previously been used in this context. Although the extension to nonmonotonic logic is fairly straightforward, an additional step is required in the abduction process: when closing the open branches of a tableau by adding literals (abducibles) one has to check that no branches that were previously closed become open as a result of extending the theory. The other main difference from approaches such as (Cialdea Mayer & Pirri 1993) and (Aliseda 1998) is that we work here in a nonclassical underlying logic.

## Equilibrium logic

Equilibrium logic is a formal system of nonmonotonic reasoning proposed and discussed in (Pearce 1997; 1999). It is currently defined for propositional logic and can therefore be applied also to grounded (quantifier-free) theories in a first-order predicate language. One of the interesting features of equilibrium logic is that it generalises the stable model and answer set semantics for logic programs, as developed in (Gelfond & Lifschitz 1988; 1990; 1991). In fact the equilibrium models of a



[1] Information about the systems `dlv`, `smodels` and others can be found in `http://www.uni-koblenz.de/ag-ki/LP/lp_systems.html`

theory coincide with its stable models or answer sets if the theory in question has the syntactic form of a logic program (for which the latter models are defined). It therefore offers a means to extend the reasoning mechanism associated with answer sets beyond the syntactic limitations of normal and disjunctive logic programs.

Equilibrium logic is based on a 5-valued logic called *here-and-there with strong negation* and denoted here by $\mathbf{N}_5$. This logic can be viewed both as an axiomatic extension of Nelson's constructive logic with strong negation, $\mathbf{N}$ (Nelson 1949), and as a conservative extension of the Smetanich logic of here-and-there (Smetanich 1960), obtained by adding the strong negation operator together with the well-known axioms of Vorob'ev, (Vorob'ev 1952a; 1952b).

Formulas are built-up using the logical constants of intuitionistic logic $\mathbf{H}$: $\wedge, \vee, \neg, \rightarrow$ together with the additional operator '$\sim$' called strong negation. The many-valued semantics uses the elements of $\mathbf{5} = \{-2,-1,0,1,2\}$ as truth values and the connectives are interpreted in the Nelson algebra, ie. $\vee$ is the maximum, $\wedge$ is the minimum and the truth tables for the connectives $\rightarrow$, $\neg$ and $\sim$ are:

| $\rightarrow$ | $-2$ | $-1$ | $0$ | $1$ | $2$ |
|---|---|---|---|---|---|
| $-2$ | 2 | 2 | 2 | 2 | 2 |
| $-1$ | 2 | 2 | 2 | 2 | 2 |
| $0$ | 2 | 2 | 2 | 2 | 2 |
| $1$ | $-2$ | $-1$ | 0 | 2 | 2 |
| $2$ | $-2$ | $-1$ | 0 | 1 | 2 |

| | $\neg$ |
|---|---|
| $-2$ | 2 |
| $-1$ | 2 |
| $0$ | 2 |
| $1$ | $-1$ |
| $2$ | $-2$ |

| | $\sim$ |
|---|---|
| $-2$ | 2 |
| $-1$ | 1 |
| $0$ | 0 |
| $1$ | $-1$ |
| $2$ | $-2$ |

The notion of model, satisfiability and validity are the usual in a many-valued logics using the value 2 as designated value.

To define equilibrium entailment, we need to introduce an order $\trianglelefteq$ between models and a notion of *total* model.

**Order $\trianglelefteq$** If $\sigma_1$ and $\sigma_2$ are two models of a set of formulas $\Pi$ then $\sigma_1 \trianglelefteq \sigma_2$ iff for every propositional variable $p$ occurring in $\Pi$ the following properties hold:

1. $\sigma_1(p) = 0$ if and only if $\sigma_2(p) = 0$.
2. If $\sigma_1(p) \geq 1$, then $\sigma_1(p) \leq \sigma_2(p)$
3. If $\sigma_1(p) \leq -1$, then $\sigma_1(p) \geq \sigma_2(p)$

**Total model** A model $\sigma$ of $\Pi$ is a *total model* if $\sigma(p) \in \{-2, 0, 2\}$ for every propositional variable $p$ in $\Pi$.

**Equilibrium model** A model $\sigma$ of $\Pi$ is said to be an *equilibrium* model of $\Pi$ if it is minimal under $\trianglelefteq$ among models of $\Pi$, and it is total.

The meaning of equilibrium can be more intuitively understood as follows. Let an objective literal be an atom or strongly negated atom. Then a model can be characterised by a set of true objective literals (atoms taking values 2 or $-2$) and a set of non-false objective literals (atoms taking values 1 and $-1$). The remaining literals (having value 0) can be thought of as false. A total model is one in which all objective literals are either true or false (ie. no atom is assigned 1 or $-1$). An equilibrium model is therefore a total model such that no other model in $\mathbf{N}_5$ has the same set of false objective literals but fewer true objective literals.

Equilibrium logic is the logic determined by the equilibrium models of a theory; we define it formally in terms of a nonmonotonic entailment relation.

**Equilibrium entailment** Let $\varphi_1, \ldots, \varphi_n, \varphi$ be formulas. We define the relation $\mid\!\sim$ called *equilibrium entailment*, as follows

1. If $\Pi = \{\varphi_1, \ldots, \varphi_n\}$ has equilibrium models, then $\varphi_1, \ldots, \varphi_n \mid\!\sim \varphi$ if every equilibrium model of $\Pi$ is a model of $\varphi$ in $\mathbf{N}_5$.
2. If either $n = 0$ or $\Pi$ has no equilibrium models, then $\varphi_1, \ldots, \varphi_n \mid\!\sim \varphi$ if $\varphi_1, \ldots, \varphi_n \models \varphi$.

## The problem of abduction

Let $\Pi$ be a theory (set of formulas) and let $\varphi$ be a formula that is consistent with the theory but not entailed by it. The problem of abduction is to find a minimal set of formulas of a certain kind which, when conjoined with $\Pi$, entail $\varphi$. We suppose that entailment is defined by the nonmonotonic relation, $\mid\!\sim$, of equilibrium entailment which we regard as an extension of the monotonic consequence relation, $\models$ of $\mathbf{N}_5$. Thus the conditions of non-entailment and consistency can be formulated as:

1. $\Pi \not\mid\!\sim \varphi$
2. $\Pi \not\models \neg\varphi$

Notice that defining consistency by $\Pi \not\mid\!\sim \neg\varphi$ is in general too strong in the context of nonmonotonic consequence since we may have $\Pi \mid\!\sim \neg\varphi$ and yet still find a consistent extension of $\Pi$ that derives $\varphi$.

It is often assumed in abductive reasoning that the abducible formulas are syntactically simple, eg. atoms or literals of a certain kind. We shall also make this assumption here, and accordingly an abductive explanation of $\varphi$ from theory $\Pi$ is defined as a formula $\alpha$ such that

1. $\Pi \cup \{\alpha\} \mid\!\sim \varphi$
2. $\Pi \cup \{\alpha\}$ is consistent
3. $\alpha$ is minimal
4. The form of $\alpha$ is syntactically restricted: conjunctions of literals of three kinds explained below

Note that in this paper we discuss abduction in the setting of sceptical reasoning (from all models). Sometimes in abductive logic programming (see eg. (Sakama & Inoue 1999)) one is interested in credulous reasoning (from a single model); the methods described can easily be modified to handle this case.

## Tableaux for equilibrium entailment

The tableau system we use here for computing equilibrium entailment was introduced in (Pearce, de Guzmán,

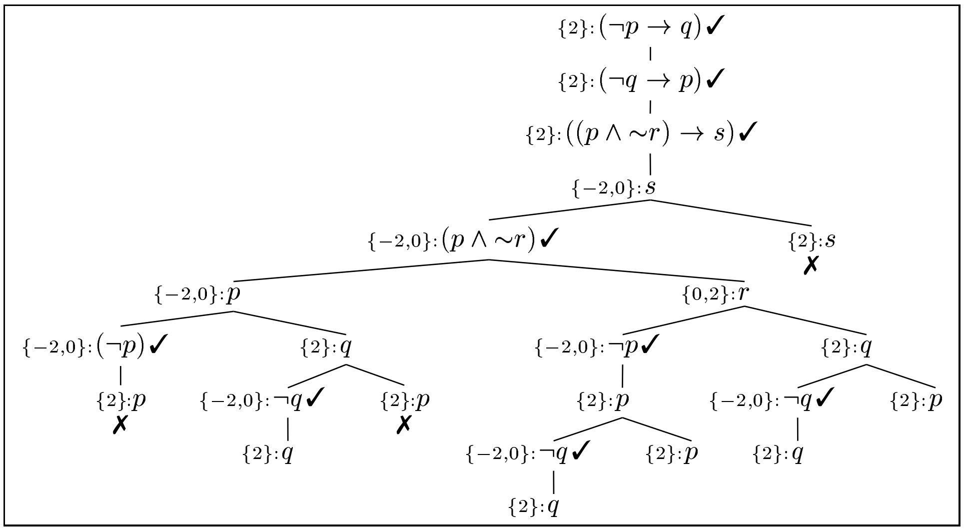


Figure 1: Main tableau for $\neg p \to q, \neg q \to p, (p \wedge \sim r) \to s \mid\!\sim s$; 5 open branches and 24 total counter-models

& Valverde 2000b) and for reasons of space we recall just its basic working principles.

Briefly, we can say that the aim of a tableau system is to generate a counter-model for the inference we wish to prove, or else conclude that no such counter-model exists and therefore that the inference in question is valid. The systems are presented by describing the construction of an initial tableau and rules for its expansion. In the initial tableau one collects the conditions that are needed to verify the required counter-model and the aim of the expansion rules is to transmit these conditions to the subformulas until the propositional variables are reached and the countermodel can be completely defined.

In the case of equilibrium entailment we use tableaux with *signed formulas*, ie. formulas labelled with a set of truth-values, the set in which the formula is to be evaluated.

**Initial Tableau**: the initial tableau for $\varphi_1, \ldots, \varphi_n \mid\!\sim \varphi$ is

$$\begin{array}{c} \{2\}{:}\varphi_1 \\ \cdots \\ \{2\}{:}\varphi_n \\ \hline \{-2,0\}{:}\varphi \end{array}$$

One reads this tableau as follows: we search for assignments giving the value 2 to the formulas $\varphi_i$ and other values to $\varphi$; in particular, given that we are interested in total models, such assignments can only allot $\varphi$ the values $-2$ and 0.

**Expansion rules**: Given the initial tableau, the expansion rules transmit the signs to the subformulas in order to define the searched model. For example, the expansion rule associated with an implication with sign $\{2\}$ is the following

$$\begin{array}{c|c} \multicolumn{2}{c}{\{2\}{:}\varphi \to \psi} \\ \hline \{-2,0\}{:}\varphi & \{2\}{:}\psi \end{array}$$

One can read this rule as follows: the formula $\varphi \to \psi$ is evaluated as 2 if $\varphi$ is evaluated as $-2$, or $\varphi$ is evaluated as 0, or else $\psi$ is evaluated as 2. As the example shows, one cannot describe the system using only the sign $\{2\}$, in particular, one needs expansion rules for every connective and for the signs $\{2\}$, $\{0,2\}$, $\{-2,0\}$, $\{-2\}$.

Once all formulas and subformulas have been expanded, the *signed literals* (ie. signed formulas of the form $S{:}p$ with $p$ a propositional variable) of each branch allows one to describe a model; a branch is called *closed* if the description of the model is inconsistent or if the model described is not in equilibrium. The former condition is the usual one for all tableau systems, while the latter is specific to equilibrium entailment. Formally, a branch is said to be *closed* if one of the following conditions is verified

1. If $S_1{:}p, \ldots, S_m{:}p$, are literals with variable $p$ in the branch and $S_1 \cap \cdots \cap S_m = \varnothing$
2. No model associated with the branch is in equilibrium.

An important characteristic of this system is that to verify the property of equilibrium one uses *auxiliary tableaux*. More precisely, for each model $\sigma$, a tableau for $\Pi$ is constructed with the aim of finding another model $\tau$, such that $\tau \trianglelefteq \sigma$ and $\tau \neq \sigma$. To describe the construction of these auxiliary tableaux these specific restrictions on the model are taken account by means of the process of *signing-up* which partially evaluates subformulas using the restrictions before deciding which expansion rule to apply. Specifically:

- If $\sigma(p) = 2$, then the model $\tau$ we are looking for must verify $\tau(p) \in \{1,2\}$
- If $\sigma(q) = -2$, then the model $\tau$ we are looking for must verify $\tau(q) \in \{-2,-1\}$.

- If $\sigma(r) = 0$, then the model $\tau$ we are looking for must verify $\tau(r) = 0$

So we have the following characterization of equilibrium entailment using the tableau system:

> $\Pi \mathrel{|\!\sim} \varphi$ if and only if there exists a closed tableaux for $\Pi$ and $\varphi$; that is, if there is no closed auxiliary tableau for a counter-model generated from the main tableau.

When the set $\Pi$ has no equilibrium models, $\mathrel{|\!\sim}$ is defined just as $\models$; in this case we need another tableau system, see (Pearce, de Guzmán, & Valverde 2000b).

### Example

Let us consider de theory $\Pi = \{\neg p \to q, \neg q \to p, (p \wedge {\sim} r) \to s\}$. This theory does not entail $s$. From the open branches in the terminated tableau for this entailment (figure 1) we obtain 24 total counter-models and two of them, $\sigma_1$ and $\sigma_2$, are equibrium models of $\Pi$, therefore, the entailment does not hold.

| | $p$ | $q$ | $r$ | $s$ |
|---|---|---|---|---|
| $\sigma_1$ | 2 | 0 | 0 | 0 |
| $\sigma_2$ | 0 | 2 | 0 | 0 |

We only include the auxiliary tableau for these counter-models in the figures 2 and 3.

$$
\begin{array}{c}
\{2\}{:}(\neg p \to q)_{\{2\}}\ \checkmark \\
| \\
\{2\}{:}(\neg q \to p)_{\{1,2\}}\ \checkmark \\
| \\
\{2\}{:}((p \wedge {\sim} r) \to s)_{\{2\}}\ \checkmark \\
| \\
\{2\}{:}((\neg q)_{\{2\}} \to p_{\{1,2\}})\ \checkmark \\
\begin{array}{cc}
\{1\}{:}(\neg q)_{\{2\}}\ \checkmark & \{2\}{:}p \\
\times & \times
\end{array}
\end{array}
$$

Figure 2: Auxiliary tableau for $\sigma_1$.

$$
\begin{array}{c}
\{2\}{:}(\neg p \to q)_{\{2\}}\ \checkmark \\
| \\
\{2\}{:}(\neg q \to p)_{\{1,2\}}\ \checkmark \\
| \\
\{2\}{:}((p \wedge {\sim} r) \to s)_{\{2\}}\ \checkmark \\
| \\
\{2\}{:}((\neg q)_{\{2\}} \to p_{\{1,2\}})\ \checkmark \\
\begin{array}{cc}
\{1\}{:}(\neg q)_{\{2\}}\ \checkmark & \{2\}{:}p \\
\times & \times
\end{array}
\end{array}
$$

Figure 3: Auxiliary tableau for $\sigma_2$.

With the tableau for $\sigma_1$ we look for a model $\tau$ such that $\tau(p) \in \{0, 2\}$ and $\tau(q) = \tau(r) = \tau(s) = 0$; this information is propagated to the formulas with the signing-up process and it is presented in the subindex; this way, the unique model of $\Pi$ from the right branch is just $\sigma_1$ and this is the reason because the branch is closed, specifically it is $\sigma_1$-closed.

## The tableau system and abduction

Having a proof system based on tableaux allows one to apply techniques for generating abductive explanations for the classical logic based on this proof system. The technique consists in analysing the models generated by the complete expansions of all formulas and adding to each open branch the literals necessary to force the closure of the branch. It is sufficient to add one literal for each branch, and therefore it suffices to add the conjunction of those literals closing branches (at most one literal per branch).

In the case of equilibrium entailment we can apply the same process: add to each open branch a literal that forces its closure. However, in this case the following problems arise:

1. The tableau used is signed and therefore the literals added are signed literals; we need to *go backwards* and construct the formula of $\mathbf{N}_5$ that generates the required signed literal in the tableau.
2. Although it suffices to add literals (conjunctions of literals) to close one or more branches, we have to take account of the fact that the literals added could open a branch that was previously closed, since these literals might affect the property of equilibrium.

Given that the generation of abductive explanations is realised by "closing" branches of a tableau with signed formulas, our first step is to generate signed versions of such explanations. Later we will need to convert these signed explanations to formulas in $\mathbf{N}_5$. Retricting ourselves to (conjunctions of) literals, the three posibilities are: $\{-2\}{:}p$, $\{0\}{:}p$, $\{2\}{:}p$, since we are interested in total models of the theory $\Pi$. We are searching for formulas of $\mathbf{N}_5$ with a single propositional variable, $\ell_{-2}(p)$, $\ell_0(p)$, $\ell_2(p)$ such that:

$$
\begin{aligned}
\{2\}{:}\ell_{-2}(p) &= \{-2\}{:}p \\
\{2\}{:}\ell_{0}(p) &= \{0\}{:}p \\
\{2\}{:}\ell_{2}(p) &= \{2\}{:}p
\end{aligned}
$$

In addition, such formulas have to be minimal among those with this property. We therefore consider the following:

$$\ell_{-2}(p) = {\sim} p, \quad \ell_0(p) = \neg p \wedge \neg{\sim} p, \quad \ell_2(p) = p$$

On the other hand, we can avoid to use the abnormal formula $\neg p \wedge \neg{\sim} p$, for the literal $\{0\}{:}p$ because we can always to decide between to weak versions $\{-2,0\}{:}p$ or $\{0,2\}{:}p$. Following the previous argument, we need two literals $\ell_{-2,0}(p)$, $\ell_{0,2}(p)$ such that:

$$
\begin{aligned}
\{2\}{:}\ell_{-2,0}(p) &= \{-2,0\}{:}p \\
\{2\}{:}\ell_{0,2}(p) &= \{0,2\}{:}p
\end{aligned}
$$

These literals are:

$$\{2\}{:}\ell_{-2,0}(p) = \neg p \qquad \{2\}{:}\ell_{0,2}(p) = \neg{\sim} p$$

### Generating explanations

The branches of a tableau that are complete for an entailment $\Pi \mathrel{|\!\sim} \varphi$ are divided into three types:

- Branches that are closed through having inconsistent sets of literals. Such branches can be ignored since their characteristics cannot be changed by adding new literals.

- Branches that are closed because all the auxiliary tableaux are open. Such branches could open upon the addition of new literals since some auxiliary tableau might close.
- Branches that are open because some auxiliary tableau is closed. Such branches can close by adding a literal, but only by contradiction because a closed auxiliary tableau cannot become open. This is an advantageous characteristic of the system: the form in which one forces closure of a branch is the same as that for classical logic.

The process of generating explanations can therefore be viewed as an iterative process briefly described as follows:

1. Construct conjunctions of literals that close open branches, these formulas are the *initial explanations*. The associated signed literals are added to the branches in the main tableau and they are analysed in order to complete the explanations.
2. If some branch becomes close by contradiction, the branch is ignored.
3. If some branch becomes open because the counter-model from the branch is now in equilibrium, then we need to close (via contradiction) the branch, by adding new literals.

If in this process the explanation becomes inconsistent, it is rejected, otherwise, the final result is an adbuctive explanation according to equilibrium entailment.

In the next section we describe this process in more detail.

## Description of the algorithm

In order to simplify the description, we are going to identify assignments in $\mathbf{N}_5$ with sets of signed literals. Specifically, if $\sigma$ is an assignment for $\Pi$ and $\{p_1, \ldots, p_n\}$ is the set of propositional variables in $\Pi$, then we identify $\sigma$ with

$$\{ {\scriptstyle\{\sigma(p_1)\}}{:}p_1, \ldots, {\scriptstyle\{\sigma(p_n)\}}{:}p_n \}$$

On the other hand, in this section we understand that the explanations are signed literals, this literal will be translated to formulas in $\mathbf{N}_5$ using the correspondence explained in the previous section.

The terminology of *closure* is taken from (Cialdea Mayer & Pirri 1993) and it is extended to $\mathbf{N}_5$ in this work. In the following, we assume that we have an open and terminated tableau (ie every formula in every branch has been expanded and some branch is open) and the equilibrium models have been generated from it.

**Initial closures** An *initial closure* is a set of signed literals $C = \{{\scriptstyle\{v_1\}}{:}p_1, \ldots, {\scriptstyle\{v_n\}}{:}p_n\}$ such that for every equilibrium counter-model, $\sigma$, there exists a literal ${\scriptstyle\{v_i\}}{:}p_i \in C$ such that ${\scriptstyle\{v_i\}}{:}p_i \notin \sigma$ and, in addition, $C$ is minimal wrt set inclusion.

An initial closure need not be an abduction explanation, and it can happen that no extension of the initial closure leads to an abductive explanation. At the same time it is also possible that several abduction explanations can be constructed that extend a given initial closure. Then for every initial closure, $C$, we want to construct a set $I_\alpha$ containing all possible abduction explanations extending $C$ ($I_\alpha$ can be empty). The construction of $I_\alpha$ is done recursively.

1. Let $I_0 = \{C\}$
2. From $I_{k-1} = \{A_1, \ldots, A_r\}$ we want to construct $I_k$. We are going to define a sequence of sets, $J_0, \ldots, J_r$ such that $J_r$ is the set $I_k$ we are looking for.
   (a) $J_0 = I_{k-1}$.
   (b) From $J_{s-1}$ we want to construct $J_s$. Let $\{\sigma_{s_1}, \ldots, \sigma_{s_\beta}\}$ be the non-equilibrium models such that $A_s \cup \sigma_{s_j}$ is in equilibrium. Then
   $$J_s = J_{s-1} \cup \{A_s \cup \sigma \mid \begin{array}{l} \sigma \cup \sigma_{s_j} \text{ has just one pair} \\ \quad \text{of contradictory literals} \\ \quad \text{for all } j \text{ and} \\ A_s \cup \sigma \text{ is consistent } \end{array}\}$$

   We define $I_k$ as $J_r$.

The set of abductive explanations extending $C$ is the set $I_\alpha$ such that $I_\alpha = I_{\alpha+1}$.

The algorithm works in the following way. For an initial closure $C$ we search for those counter-models that were previously not in equilibrium and now are in equilibrium if we add the explanation $C$. These counter-models are cancelled by adding new literals using the construction of the sets $J_s$. In general there will be different ways to produce this extension and therefore each one must be submitted to the same process. During the process of extension, those explanations that are inconsistent are eliminated, and the process finishes when the last extension doesn't alter the equilibrium property of any model.

In practice, the algorithm works by handling all tableaux, main and auxiliary ones. Studying the counter-models, we generate partial closures and these closures are added to the open branches in the tableaux in order to update them; so it is no necesary to construct other tableaux in the recursive steps of the algorithm.

### Continuing the example

In the example from the previous page we saw that $\Pi = \{\neg p \to q, \neg q \to p, (p \wedge {\sim} r) \to s\}$ does not entail $s$, now we want to generate explanations for $s$ extending $\Pi$.

- First we generate the initial closures, that is the sets of literals that cancel the counter-models $\sigma_1$ and $\sigma_2$: $\{{\scriptstyle\{-2\}}{:}p\}$, $\{{\scriptstyle\{-2\}}{:}q\}$, $\{{\scriptstyle\{-2\}}{:}r\}$, $\{{\scriptstyle\{2\}}{:}r\}$ (we dont consider either $\{{\scriptstyle\{2\}}{:}s\}$, because it is a trivial explanation, or $\{{\scriptstyle\{-2\}}{:}s\}$, because it only can be extended with inconsistency).

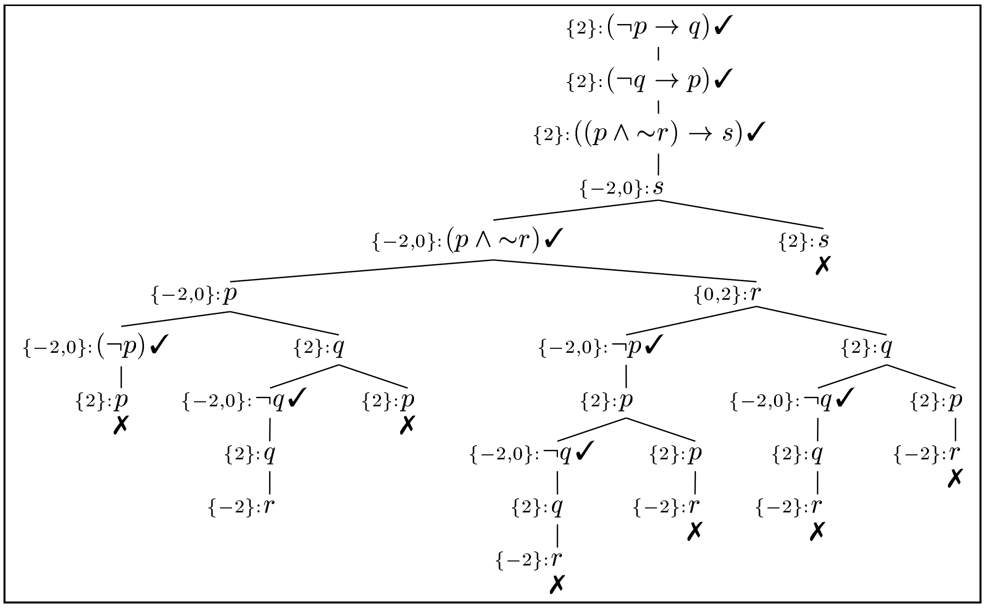


Figure 4: Main tableau for $\neg p \to q, \neg q \to p, (p \wedge \sim r) \to s \mathrel{|\!\sim} s$ updated with $\{-2\}{:}r$; it has one open branch and generates 4 countermodels

To continue with the example, we are going to extend the initial explanation $\{\{-2\}{:}r\}$.

- Adding this literal to the main tableau we obtain the updated tableau in figure 4 where several branches become closed and the number of total counter-models decreases, we have now 4 counter-models:

| | $p$ | $q$ | $r$ | $s$ |
|---|---|---|---|---|
| $\sigma_3$ | 2 | −2 | −2 | 0 |
| $\sigma_4$ | 0 | 2 | −2 | 0 |
| $\sigma_5$ | 2 | −2 | −2 | −2 |
| $\sigma_6$ | 0 | 2 | −2 | −2 |

  Updating the auxiliary tableaux for these models we obtain that $\sigma_4$ is an equilibrium model of $\Pi \cup \{\{-2\}{:}r\}$, that is, this extension does not entail $s$.

- Continuing with the algorithm, we need to extend $\Pi \cup \{\{-2\}{:}r\}$ adding literals cancelling the counter-model $\sigma_4$. We have several possibilities:
  - Adding $\{0\}{:}q$, all the branches in the main tableau become closed, then the resulting extension is an aceptable explanation if the resulting theory has equilibrium models. In this case, we can add the signed literal $\{-2,0\}{:}q$ and then the explanations are $\{\sim r, \neg q\}$. A final check is needed to see if the resulting theory $\Pi \cup \{\sim r, \neg q\}$ has equilibrium models; using a new tableau we obtain that $\tau(p) = \tau(s) = 2$, $\tau(q) = 0$, $\tau(r) = -2$ is an equilibrium model.
  - Adding $\{2\}{:}p$ all the branches become also closed. The corresponding extension $\{\sim r, p\}$ is an explanation if $\Pi \cup \{\sim r, p\}$ has equilibrium models; the model $\tau$ in the previous item is also an equilibrium model for $\Pi \cup \{\sim r, p\}$
  - Adding $\{-2\}{:}p$, some branches stay open in the main tableau and we have two total counter-models:

| | $p$ | $q$ | $r$ | $s$ |
|---|---|---|---|---|
| $\sigma_7$ | −2 | 2 | −2 | 0 |
| $\sigma_8$ | −2 | 2 | −2 | −2 |

    Updating the auxiliary tableaux, we obtain that $\sigma_7$ is an equilibrium model of $\Pi \cup \{\{-2\}{:}r, \{-2\}{:}p\}$ and some additional extension is needed, but no other extension produces a theory with equilibrium models and thus this explanation is ignored.

Therefore, we have found two abductive explanations extending $\{\sim r\}$:

$$\{\sim r, \neg q\} \qquad \{\sim r, p\}$$

The other initial closures can not be extended to non-trivial or consistent explanations and thus these are the unique abductive explanations.

## Future work

The algorithm presented does not yet take account of conditions of *minimality*. (Note that also `dlv` does not support the generation of abducibles in disjunctive programs if minimality conditions are considered.) Although we have not analysed such conditions in the present work, such an analysis would not alter the basic ideas of the present algorithm. Adding consideration of minimality may however help improve the efficiency of the algorithm; this topic will be considered in future work.

Another important question that remains open is the complexity of the general inference problem. Some preliminary results have been obtained but our study of this issue is still ongoing.

Finally, we still need to consider the generation of consistent extensions of a theory without equilibrium models, but also explaining the required conclusion. For this we need to work with a tableau system for the relation $\models$ in $\mathbf{N}_5$ in order to generate consistent extensions of $\Pi$ without equilibrium models but infering the conclusion. We plan develop a system to generate abductives in $\mathbf{N}_5$ in a monotonic sense.

## Concluding remarks

We have sketched a method for generating abductive explanations in the framework of equilibrium logic, a system generalising answer set semantics beyond the syntax of normal and disjunctive logic programs. The method is based on tableaux systems for checking entailment in equilibrium logic and extends previous work on abduction using tableaux, notably that of (Cialdea Mayer & Pirri 1993; Palopoli, Pirri, & Pizzuti 1999; Aliseda 1998). Since equilibrium logic conservatively extends (the inference mechanism of) answer set semantics, the methods outlined here should be of interest for those wishing to explore knowledge representation and reasoning issues beyond the current syntactic limitations of answer set programming, eg. to programs with nested expressions in the style of (Lifschitz, Tang, & Turner 1999), and beyond. While they are relatively efficient on ordinary logic programs, it is unclear how the current generation of answer set solvers could be easily modified to embrace more recent extensions of the concept of answer set. Tableau systems on the other hand are based from the outset on an unrestricted (propositional) syntax and are therefore suitable for handling such more extensive classes of formulas and theories.

## References


Aliseda, A. 1998. Computing abduction in semantic tableaux. *Revista Iberoamericana de Computación* II(1):5–13.

Cialdea Mayer, M., and Pirri, F. 1993. First-order abduction via tableau and sequent calculi. *Bulletin of the IPGL* 1(1):99–117.

Gelfond, M., and Lifschitz, V. 1988. The stable semantics for logic programs. In Kowalski, R., and Bowen, K., eds., *Proceedings of the 5th international symposium on logic programming*, 1070–1080. Cambridge, MA.: MIT Press.

Gelfond, M., and Lifschitz, V. 1990. Logic Program with Classical Negation. In Warren, D., and Szeredi, P., eds., *Proceedings of the 7th Int. Conf. on Logic Programming*, 579–597. MIT.

Gelfond, M., and Lifschitz, V. 1991. Classical negation in logic programs and disjunctive databases. *New Generation Computing* 9(3–4):365–385.

Kakas, A. C., and Michael, A. 1998. Applications of abductive logic programming. In Jaffar, J., ed., *Proceedings of the 1998 Joint International Conference and Symposium on Logic Programming (JICSLP-98)*, 343–344. Cambridge: MIT Press.

Kakas, A.; Kowalski, R.; and Toni, F. 1993. Abductive logic programming. *Journal of Logic and Computation* 6?(2?):719–770.

Lifschitz, V.; Pearce, D.; and Valverde, A. 2000. Strong equivalent logic programs. *Transactions on Computational Logic* (to appear).

Lifschitz, V.; Tang, L. R.; and Turner, H. 1999. Nested expressions in logic programs. *Annals of Mathematics and Artificial Intelligence* 25(3–4):369–389.

Nelson, D. 1949. Constructible falsity. *Journal of Symbolic Logic* 14(2):16–26.

Palopoli, L.; Pirri, F.; and Pizzuti, C. 1999. Algorithms for selective enumeration of prime implicants. *Artificial Intelligence* 111(1–2):41–72.

Pearce, D.; de Guzmán, I. P.; and Valverde, A. 2000a. Computing equilibrium models using signed formulas. In *First International Conference on Computational Logic, CL 2000, LNAI 1861*, 688–702. London, UK: Springer-Verlag.

Pearce, D.; de Guzmán, I. P.; and Valverde, A. 2000b. A tableau calculus for equilibrium entailment. In Dyckhoff, R., ed., *Automated Reasoning with Analytic Tableaux and Related Methods, TABLEAUX 2000, LNAI 1847*, 352–367. St. Andrews, Scotland: Springer-Verlag.

Pearce, D. 1997. A new logical characterisation of stable models and answer sets. In Dix, J.; Pereira, L.; and Przymusinski, T., eds., *Non-monotonic Extensions of Logic Programming, NMELP'96, LNAI 1216*, 57–70. Springer-Verlag.

Pearce, D. 1999. From here to there: Stable negation in logic programming. In Gabbay, D., and Wansing, H., eds., *What is Negation?* Kluwer.

Sakama, C., and Inoue, K. 1999. Updating extended logic programs through abduction. In Gelfond, M.; Leone, N.; and Pfeifer, G., eds., *Proceedings of the 5th International Conference on Logic Programming and Nonmonotonic Reasoning (LPNMR-99)*, volume 1730 of *LNAI*, 147–161. Berlin: Springer.

Smetanich, Ya, S. 1960. On completeness of a propositional calculus with an additional operation of one variable (in russian). *Trudy Moscovskogo matrmaticheskogo obshchestva* 9:357–372.

Vorob'ev, N. N. 1952a. A constructive propositional calculus with strong negation (in russian). *Doklady Akademii Nauk SSR* 85:465–468.

Vorob'ev, N. N. 1952b. The problem of deducibility in constructive propositional calculus with strong negation (in russian). *Doklady Akademii Nauk SSR* 85:689–692.