ELSEVIER

Artificial Intelligence 93 (1997) 63-101

Artificial Intelligence

An abstract, argumentation-theoretic approach to default reasoning

A. Bondarenko[a,1], P.M. Dung[b,*], R.A. Kowalski[c,2], F. Toni[c,3]

[a] *Programming Systems Institute, Russian Academy of Sciences, Pereslavle-Zalessky, Russia*

[b] *Computer Science Department, Asian Institute of Technology, P.O. Box 2754, Bangkok 10501, Thailand*

[c] *Department of Computing, Imperial College, 180 Queen's Gate, London SW7 2BZ, United Kingdom*

Received 13 December 1995

Abstract

We present an abstract framework for default reasoning, which includes Theorist, default logic, logic programming, autoepistemic logic, non-monotonic modal logics, and certain instances of circumscription as special cases. The framework can be understood as a generalisation of Theorist. The generalisation allows any theory formulated in a monotonic logic to be extended by a defeasible set of assumptions.

An assumption can be defeated (or "attacked") if its "contrary" can be proved, possibly with the aid of other conflicting assumptions. We show that, given such a framework, the standard semantics of most logics for default reasoning can be understood as sanctioning a set of assumptions, as an extension of a given theory, if and only if the set of assumptions is conflict-free (in the sense that it does not attack itself) and it attacks every assumption not in the set.

We propose a more liberal, argumentation-theoretic semantics, based upon the notion of admissible extension in logic programming. We regard a set of assumptions, in general, as admissible if and only if it is conflict-free and defends itself (by attacking) every set of assumptions which attacks it. We identify conditions for the existence of extensions and for the equivalence of different semantics. © 1997 Elsevier Science B.V.

Keywords: Default reasoning; Nonmonotonic reasoning; Abduction; Argumentation; Nonmonotonic logics; Default logic; Autoepistemic logic; Nonmonotonic modal logics; Circumscription; Logic programming; Theorist

[*] Corresponding author. E-mail: dung@cs.ait.ac.th.

[1] E-mail: andrei@troyka.msk.su.

[2] E-mail: rak@doc.ic.ac.uk.

[3] E-mail: ft@doc.ic.ac.uk.

0004-3702/97/$17.00 © 1997 Elsevier Science B.V. All rights reserved.

PII S0004-3702(97)00015-5

1. Introduction

Until recently, formal logic was concerned mainly with the formalisation of universal "truths", such as those of mathematics, which hold without exception and for all time. The logics which have proved useful for this purpose are all *monotonic*, in the sense that any logical consequence of a set of axioms remains a logical consequence if new axioms are added. Because of the default character of human reasoning, that certain beliefs hold by default if there is no reason to believe the contrary, attempts to apply such monotonic logics to the formalisation of human reasoning have met with limited success. For this reason a number of "non-monotonic" logics [38-40,49] have been developed.

In this paper, we show that many of these logics can be understood as special cases of a single abstract framework, based upon an argumentation-theoretic interpretation of the semantics of logic programming [16,17] and its abstractions [6,10,11,27]. In this framework, a set of assumptions, formulated in an underlying monotonic logic, is regarded as an acceptable extension of a given theory, unless and until there is reason to believe some contrary set of assumptions. Non-monotonicity arises because the addition of a new sentence to a theory may provide new evidence to the contrary of a previously acceptable default conclusion, which now has to be withdrawn.

We show that the standard semantics associated with most non-monotonic logics imposes a further requirement for the acceptability of a set of assumptions, namely that the set attacks every other assumption not in the set. (A set of assumptions attacks an assumption if and only if together with the given theory it implies a sentence contrary to the assumption in the underlying monotonic logic.) The following simple example illustrates informally the way in which various non-monotonic logics can be viewed as instances of the same abstract framework.

Example 1.1. Consider the principle that

A person is innocent unless proved guilty.

Its informal English meaning is that if a person is accused of a crime, then the burden of proof is on the prosecution to show that the accused is guilty, rather than on the defence to show that he is not. The accused is assumed not guilty, by default, unless the contrary can be shown.

The naive representation in classical logic

$$\forall X[\neg guilty(X) \rightarrow innocent(X)]$$

fails to capture the default character of the principle. It imposes on the defence the greater burden of explicitly establishing that the accused is not guilty. In general, this will be harder than simply showing there is no proof that he is guilty. In particular, in the commonly occurring case where there is insufficient evidence to prove either that the accused is guilty or that he is not, the default principle gives the accused the benefit of doubt and concludes that he is innocent. In contrast, the representation in classical logic fails to imply any conclusion.

In classical logic, the naive representation logically implies the contrapositive

$$\forall X[\neg innocent(X) \rightarrow guilty(X)]$$

and therefore treats innocence and guilt equally. The informal principle, however, expresses that innocence, rather than guilt, holds by default. Default reasoning is non-monotonic, because a conclusion (e.g. that a person is innocent) which is justified in one state of knowledge may not be justified if new knowledge becomes available.

Theorist [43] employs the "naive representation" of classical logic, but overcomes its deficiencies by extending the theory which includes the given sentence by means of a maximal consistent set of assumptions of the form

$$\neg guilty(t)$$

for all *ground* (i.e. variable-free) terms, *t*, of the language. The asymmetric character of the default is captured by not considering extensions with assumptions of the form

$$\neg innocent(t).$$

Because of this selective use of assumptions, the use of the contrapositive in this example does not give rise to unintended consequences.

Like Theorist, circumscription [38] also employs the "naive representation" of classical logic and minimises the extension of the predicate *guilty* (because minimising positive instances is equivalent to maximising negative instances of a predicate). Although Theorist views extensions as syntactic objects and circumscription views them as model-theoretic, the two views are equivalent in many cases.

Theorist and circumscription differ in another respect. Theorist is *credulous*, in that it sanctions holding a conclusion if it is a logical consequence of one maximal consistent extension of the given theory, whereas circumscription is *sceptical*, in its sanctioning a conclusion if it holds in all such extensions (more precisely, if it holds in all minimal models).[4]

Logic programming can be understood, similarly to Theorist, as extending theories by means of ground negative literals *not p* representing the assumption that *not p* holds by default unless its *contrary*, *p*, can be shown. Thus the logic programming representation

$$innocent(X) \leftarrow not\ guilty(X)$$

can be understood as expressing literally that a person is innocent if the person can not be proved guilty; or equivalently, in our framework, as expressing that a person is innocent if the contrary of the assumption that the person is not guilty can not be shown.

Logic programming considers all ground negative literals as possible assumptions, but prevents the derivation of the contrapositive

$$q \leftarrow not\ p$$

[4] However, Poole [44] has also proposed an extension of Theorist, in which credulous reasoning is used for "explanation" and sceptical reasoning for "prediction".

of

$$p \leftarrow not\ q$$

by employing only modus ponens for the implication symbol, "$\leftarrow$".[5] Together with instantiation of universally quantified variables, these two inference rules constitute the underlying monotonic logic upon which logic programming is based. As we shall see later, many different credulous and sceptical semantics for logic programming can be understood in such assumption-based terms.

Default logic [49] combines classical logic with domain-specific inference rules. In this example, it might employ the representation

$$\frac{M\neg guilty(X)}{innocent(X)}$$

where Mp stands for "p is consistent", i.e. the *contrary*, $\neg p$, cannot be shown, where "$\neg$" is classical negation. Thus the domain-specific inference rule can be interpreted as expressing that a person can be shown to be innocent if the contrary of the assumption that the person is not guilty cannot be shown. In our framework, this is very similar to the interpretation of the logic programming representation.

Like logic programming, default logic prevents the derivation of the contrapositive of default rules. A domain-specific inference rule of the form

$$\frac{M\neg p}{q}$$

can be used to derive q. It does not sanction the "contrapositive inference rule"

$$\frac{M\neg q}{p}$$

In our framework, default logic can be understood as non-monotonically adding assumptions of the form Mp to theories formulated in an underlying monotonic logic, which consists of classical logic augmented with domain specific inference rules. We will see later that the standard semantics of default logic can be understood as a credulous semantics in assumption-based terms.

Autoepistemic logic [40] and non-monotonic modal logics [39], on the other hand, can both be understood as using an expression of the form $\neg Lp$ to represent an assumption which holds by default if the *contrary*, namely p, cannot be shown. "L" is a modal operator, meaning "is believed", is "known" or "can be shown". "$\neg$", as in default logic, is classical negation. Thus, in both autoepistemic and nonmonotonic modal logic, the example can be represented in the form[6]

$$\forall X[\neg Lguilty(X) \rightarrow innocent(X)]$$

where "$\rightarrow$" is ordinary material implication.

[5] This is equivalent to treating the implication $p \leftarrow not\ q$ as an inference rule $\frac{not\ q}{p}$.

[6] Although the form of autoepistemic logic introduced in [40] was propositional, in this paper we follow subsequent first-order formulations.

Both logics allow the derivation of the contrapositive

$$\neg p \rightarrow Lq$$

of an implication of the form

$$\neg Lq \rightarrow p.$$

In our example, the contrapositive means that if a person is not innocent then he must be shown to be guilty, which is compatible with the default interpretation of the original sentence.

Both these logics can be understood as non-monotonically adding assumptions of the form $\neg Lp$ to theories expressed in an underlying monotonic logic. In the case of autoepistemic logic the underlying logic is classical logic, and additional assumptions of the form Lp also need to be considered explicitly. In non-monotonic modal logics, the underlying logic is modal logic, which, because it includes the necessitation rule of inference

$$\frac{p}{Lp}$$

obviates the need to consider explicit assumption of the form Lp. In both cases, the standard semantics can be understood as special cases of a single, abstract, credulous semantics, formulated in assumption-based terms, which includes the stable model semantics [20] of logic programming and the standard semantics of default logic as further special cases.

The "innocent-unless-guilty" example illustrates the common feature of all these non-monotonic logics, namely that they can be understood as adding assumptions to an underlying monotonic logic, provided the contrary cannot be shown. In the general case, however, the problem of showing that a sentence p cannot be shown is complicated by the fact that the attempt to show p can make use of other conflicting assumptions. Thus, for example, it is possible to have two conflicting defaults:

a person is innocent if not proved guilty,

a person is guilty if not proved innocent

or even a single conflicting default

a person is innocent if not proved innocent.

It is the need to deal with such examples that accounts for much of the complexity of non-monotonic logics.

In this paper we will investigate both credulous and sceptical ways of understanding what it means for a given conclusion to hold non-monotonically as a result of making certain default assumptions. The credulous approach justifies holding a conclusion if there is a suitably acceptable set of assumptions, extending the initial theory, from which the conclusion can be derived in the underlying monotonic logic. The sceptical approach, on the other hand, justifies a conclusion if it can be derived from *all* acceptable extensions of the given theory. The notion of acceptable extension can be understood in several different ways.

A *semantics* for default reasoning is given by specifying the notion of acceptable extension and identifying whether the approach is credulous or sceptical.

The simplest notion of acceptability, which in its credulous manifestation we call the *naive semantics*, requires simply that the initial theory be extended with some maximal set of assumptions which is *conflict-free* (in the sense that the contrary of none of the assumptions in the set can be shown using the notion of consequence in the underlying monotonic logic). This semantics generalises the semantics of Theorist [43], in which the underlying logic is classical, first-order logic.

The second, credulous semantics generalises the stable model semantics of logic programming and the standard semantics of default logic, autoepistemic logic and non-monotonic modal logic. This semantics, which we call the *stable semantics*, requires not only that an acceptable set of assumptions be conflict-free, but that, together with the initial theory, it implies the contrary of all assumptions not contained in the deductive closure of the set.

The stable semantics can be given an argumentation-theoretic interpretation, which suggests other, improved semantics. We interpret a monotonic proof of the contrary of an assumption α based upon an initial theory T extended with assumptions Δ as an *argument* against α. Abstracting away from the detail of the actual argument and focussing instead on the assumptions Δ upon which the argument is based, we say that Δ *attacks* α. Under this interpretation, a set of assumptions is *stable* if and only if it does not attack itself (i.e. is conflict-free) and attacks every assumption it does not contain.

Viewed in such argumentation-theoretic terms, stable semantics is unnecessarily opinionated, taking a stand on every issue (i.e. every possible assumption either belongs to a stable set or is attacked by it), whether or not that issue is relevant to a given conclusion under consideration. The third, credulous semantics, instead, regards a set of assumptions as acceptable if and only if it is conflict-free and its deductive closure *defends* itself against all attacks (by attacking all sets of assumptions which attack it). This semantics, called the *admissibility semantics*, generalises the admissibility semantics [10] of logic programming and arguably improves upon the standard stability semantics of default logic, autoepistemic logic and non-monotonic modal logics.

The fourth, credulous semantics, called the *preferential semantics*, simply regards an extension as acceptable if it is maximal admissible, in the sense that no proper superset of the extension is also admissible.

The fifth, credulous semantics, called the *complete semantics*, is intermediate between the admissibility and preferential semantics. It regards an extension as acceptable if it is admissible and it contains all assumptions it defends.

As mentioned above, each of these credulous semantics has a sceptical version. We will see that, in certain cases, circumscription can be understood as the sceptical version of the naive semantics, where, as in Theorist, the underlying monotonic logic is first-order classical logic. We will also see that the *well-founded semantics* of logic programming is the sceptical version of the complete semantics, where the underlying monotonic logic is the logic of Horn clauses.

The rest of the paper has the following structure: Section 2 introduces the abstract framework and the naive semantics, equivalent to the semantics of Theorist [43], and shows how different logics for default reasoning can be expressed as instances of the abstract framework. Section 3 investigates the stable semantics. Section 4 investigates the admissibility and preferential semantics. Section 5 investigates the complete semantics. Section 6 investigates sceptical semantics. Section 7 presents results about the existence of (credulous and sceptical) semantics and about certain conditions under which they are equivalent. Section 8 describes relationships to other argumentation-theoretic formalisms. Section 9 gives conclusions and points to some directions for future research.

2. Assumption-based frameworks and naive semantics

In this paper, a *deductive system* is a pair $(\mathcal{L}, \mathcal{R})$ where

- $\mathcal{L}$ is a formal language consisting of countably many sentences, and
- $\mathcal{R}$ is a set of inference rules of the form

$$\frac{\alpha_1, \ldots, \alpha_n}{\alpha}$$

where $\alpha, \alpha_1, \ldots, \alpha_n \in \mathcal{L}$ and $n \geqslant 0$.

Notice that logical axioms, α, can be represented as inference rules with $n = 0$. Any set of sentences $T \subseteq \mathcal{L}$ is called a *theory*.

A *deduction* from a theory T is a sequence $\beta_1, \ldots, \beta_m$, where $m > 0$, such that, for all $i = 1, \ldots, m$,

- $\beta_i \in T$, or
- there exists $\frac{\alpha_1, \ldots, \alpha_n}{\beta_i}$ in $\mathcal{R}$ such that $\alpha_1, \ldots, \alpha_n \in \{\beta_1, \ldots, \beta_{i-1}\}$.

$T \vdash \alpha$ means that there is a deduction from T whose last element is α. $Th(T)$ is the set $\{\alpha \in \mathcal{L} \mid T \vdash \alpha\}$.

Notice that, because all deductions have finite length, every deductive system $(\mathcal{L}, \mathcal{R})$ is *compact* in the sense that whenever $T \vdash \alpha$, then $T_0 \vdash \alpha$ for some finite subset T_0 of T. Notice, too, that every deductive system is *monotonic* in the sense that $T \subseteq T'$ implies $Th(T) \subseteq Th(T')$.

Following Poole [43], we argue that the non-monotonic character of default reasoning arises because a set of assumptions that acceptably extends a given theory in a monotonic logic might not be acceptable if new sentences are added to the theory. Different logics for default reasoning can be understood as having different underlying monotonic logics, different kinds of assumptions and different notions of acceptability.

At a sufficiently abstract level, however, despite these differences, the different credulous non-monotonic logics can all be viewed as sanctioning a set of assumptions as an acceptable extension of a given theory if and only if, given the extension, there is no reason to believe the contrary of any assumption in the set. The notion of the contrary of an assumption is different in different logics. In the simplest case, we can understand the contrary of an assumption α as its classical negation $\neg\alpha$. However, other notions of "contrariness" are needed in other cases.

Definition 2.1. Given a deductive system $(\mathcal{L}, \mathcal{R})$, an *assumption-based framework* with respect to $(\mathcal{L}, \mathcal{R})$ is a tuple $\langle T, Ab, \overline{\ } \rangle$, where

- $T, Ab \subseteq \mathcal{L}$ and $Ab \neq \emptyset$,
- $\overline{\ }$ is a mapping from Ab into $\mathcal{L}$, where $\overline{\alpha}$ denotes the *contrary* of α.

The theory T expresses a given set of beliefs, and Ab is a set of assumptions that can be used to extend T.

In the sequel, when there is no danger of ambiguity, we often omit reference to the underlying deductive system $(\mathcal{L}, \mathcal{R})$ and/or to the assumption-based framework $\langle T, Ab, \overline{\ } \rangle$.

In contrast with an earlier formalisation [6], we do not try to reduce the notion of contrariness to the notion of inconsistency. Nor, if the underlying logic admits the notion of inconsistency, do we assume that inconsistency implies every sentence of the language.

In this section, we consider the generalisation of Theorist's semantics, where the requirement that an extension be maximal consistent is generalised to the requirement that it be maximal conflict-free. We call this generalisation the *naive semantics.*

Definition 2.2. Given an assumption-based framework $\langle T, Ab, \overline{\ } \rangle$ and $\Delta \subseteq Ab$,

- Δ is *conflict-free* if and only if for all $\alpha \in Ab$, $T \cup \Delta \not\vdash \alpha, \overline{\alpha}$,
- Δ is *maximal conflict-free* if and only if Δ is conflict-free and there is no conflict-free $\Delta' \supset \Delta$.

Note that, given any set of assumptions Δ, we can form the deductive closure $Th(T \cup \Delta)$ of the theory $T \cup \Delta$. The deductive closure is often called an *extension* in the literature on non-monotonic logic. This use of the term "extension" differs from our informal use of the term, either to refer to Δ itself or to $T \cup \Delta$. In the sequel, whenever it is important to be precise, we will state explicitly which of these three uses of the term is intended.

The naive semantics is guaranteed to exist for assumption-based frameworks that admit at least one conflict-free extension, as implied by the following

Theorem 2.3. *For every conflict-free set of assumptions Δ, there exists a maximal conflict-free set of assumptions Δ' such that $\Delta \subseteq \Delta'$.*

Proof. Let $\alpha_0, \alpha_1, \ldots, \alpha_n, \ldots$ be an enumeration of $Ab - \Delta$. Let

- $\Delta_0 = \Delta$,
- $\Delta_{n+1} = \Delta_n \cup \{\alpha_n\}$ if $\Delta_n \cup \{\alpha_n\}$ is conflict-free, and $\Delta_{n+1} = \Delta_n$ otherwise.

Let $\Delta' = \bigcup_i \Delta_i$. Obviously, $\Delta \subseteq \Delta'$. Moreover, it is easy to see that Δ' is maximal conflict-free. □

2.1. Theorist

Given a deductive system $(\mathcal{L}, \mathcal{R})$ for classical first-order logic, an *abductive framework* [43] is a pair (T, Ab), where $T \subseteq \mathcal{L}$ is consistent and $Ab \subseteq \mathcal{L}$. A *scenario* is a consistent theory $T \cup \Delta$ where $\Delta \subseteq Ab$.[7] An *extension* $Th(T \cup \Delta)$ is the logical closure of a maximal (with respect to set inclusion) scenario [43].

The assumption-based framework corresponding to (T, Ab) is $\langle T, Ab, \overline{} \rangle$, where $\overline{\alpha} = \neg\alpha$, for each $\alpha \in Ab$.

Note that, since T is consistent, $\emptyset$ is a conflict-free set of assumptions. Therefore, by Theorem 2.3, the naive semantics of $\langle T, Ab, \overline{} \rangle$ always exists.

It follows immediately that

Theorem 2.4. *Given an abductive framework (T, Ab) and the corresponding assumption-based framework $\langle T, Ab, \overline{} \rangle$,*

- *$T \cup \Delta$ is a scenario of (T, Ab) if and only if Δ is a conflict-free set of assumptions in $\langle T, Ab, \overline{} \rangle$,*
- *E is an extension of (T, Ab) if and only if $E = Th(T \cup \Delta)$ where Δ is a maximal (with respect to set inclusion) conflict-free set of assumptions in $\langle T, Ab, \overline{} \rangle$.*

Example 2.5. A simplified, propositional representation in Theorist of the "innocent-unless-guilty" example of the Introduction is

$$T = \{\neg guilty \rightarrow innocent\}$$

$$Ab = \{\neg guilty\}$$

which has the single extension $Th(T \cup \{\neg guilty\})$.

2.2. *Logic programming*

We will assume, as is conventional, that the semantics of a logic program containing variables is given by the set of all its ground instances over the Herbrand universe corresponding to the language of the program. The *Herbrand universe* corresponding to a given language consists of all ground terms constructible from the constant symbols and function symbols of the language. We use $\mathcal{HB}$ to stand for the *Herbrand base*, i.e. the set of all ground atoms formulated in terms of the Herbrand universe. We use $\mathcal{HB}_{not}$ to stand for the set $\{not\ \alpha \mid \alpha \in \mathcal{HB}\}$ and Lit to stand for $\mathcal{HB} \cup \mathcal{HB}_{not}$.

A *normal logic program* is a set of *clauses* of the form

$$\alpha \leftarrow \beta_1, \ldots, \beta_n$$

where $\alpha \in \mathcal{HB}$, $\beta_1, \ldots, \beta_n \in Lit$, and $n \geqslant 0$.

The assumption-based framework corresponding to such a normal logic program T is $\langle T, \mathcal{HB}_{not}, \overline{} \rangle$ with respect to $(\mathcal{L}, \mathcal{R})$, where

- $\mathcal{L} = Lit \cup \{\alpha \leftarrow \beta_1, \ldots, \beta_n \mid \alpha \in \mathcal{HB}$ and $\beta_1, \ldots, \beta_n \in Lit$ and $n \geqslant 0\}$;
- $\mathcal{R}$ is the set of all inference rules of the form

[7] Poole defines Ab to be a set of open first-order formulae and Δ to be a set of variable-free instances of formulae in Ab. In our formulation, $\mathcal{L}$ and therefore Ab is a set of sentences (without free variables). Our formulation is equivalent to Poole's and more convenient for our purposes.

$$\frac{\alpha \leftarrow \beta_1, \ldots, \beta_n \quad \beta_1, \ldots, \beta_n}{\alpha}$$

where $\alpha \in \mathcal{HB}$ and $\beta_1, \ldots, \beta_n \in Lit$ and $n \geqslant 0$;

- $\overline{not\ \alpha} = \alpha$, for each $not\ \alpha \in \mathcal{HB}_{not}$.

The interpretation of negative literals as assumptions in logic programming was introduced in [16,17], and formed the basis for the admissibility semantics [10], the stable theory and acceptability semantics [28], and the argumentation-theoretic interpretation for these semantics presented in [11,25].

Note that we could, equivalently, represent clauses

$$\alpha \leftarrow \beta_1, \ldots, \beta_n$$

as inference rules

$$\frac{\beta_1, \ldots, \beta_n}{\alpha}$$

In this representation, the theory is empty, and a logic program is represented by domain-specific inference rules of the underlying deductive system. This alternative representation highlights the similarity between logic programming and default logic (see Section 2.3).

Example 2.6. The logic program T

$$\{innocent \leftarrow not\ guilty\}$$

represents the simplified "innocent-unless-guilty" example. In the corresponding assumption-based framework there are two maximal conflict-free sets of assumptions, $\Delta_1 = \{not\ guilty\}$ and $\Delta_2 = \{not\ innocent\}$. However, only the first, intuitively correct one is acceptable in all semantics for logic programming. Therefore, the naive semantics is not appropriate to capture the semantics for logic programming. In Sections 3, 4, 5 and 6, we will define other abstract semantics that correspond to the logic programming semantics.

Logic programming can be extended, as proposed by Gelfond and Lifschitz [21], by allowing, in addition to negation as failure, a second, explicit form of negation, written as $\sim$. This negation can be used to define negative instances of predicates explicitly, instead of inferring them implicitly using negation as failure. Abductive logic programming [25,26] is another extension of logic programming, where positive atoms can be explicitly indicated as assumptions and integrity constraints can be used to prevent unwanted assumptions.

Both extended and abductive logic programming can be formulated as instances of the assumption-based framework, following [1,6,12,25,26,56].

2.3. Default logic

Let $(\mathcal{L}_0, \mathcal{R}_0)$ be a deductive system for classical first-order logic. Following [49], a *default theory* is a pair (T, D) where

- $T \subseteq \mathcal{L}_0$,
- D is a set of default rules of the form[8]

$$\frac{\alpha, M\beta_1, \ldots, M\beta_n}{\gamma}$$

where $\alpha, \beta_1, \ldots, \beta_n, \gamma \in \mathcal{L}_0$, and $n \geqslant 0$.

Here, for simplicity, we have assumed that all default rules in D contain no free variables. (Similarly to the case of logic programming, default rules containing free variables can be understood as representing the set of all their ground instances.)

Given a default theory (T, D), the corresponding deductive system, $(\mathcal{L}, \mathcal{R})$, and assumption-based framework, $\langle T, Ab, \overline{}\rangle$, are defined by:

- $\mathcal{L} = \mathcal{L}_0 \cup \{M\alpha \mid \alpha \in \mathcal{L}_0\}$,
- $\mathcal{R} = \mathcal{R}_0 \cup D$,
- $Ab = \{M\beta \mid \beta \in \mathcal{L}_0$ and $M\beta$ appears in one of the default rules in $D\}$;
- $\overline{M\alpha} = \neg\alpha$.

Intuitively, an assumption of the form $M\alpha$ means that α is consistent, i.e. that $\neg\alpha$ cannot be shown.

We will assume that the inference rules of first-order logic in $\mathcal{R}_0$ are applied only to formulas in $\mathcal{L}_0$. This assumption together with the fact that the default rules in D derive only formulas in $\mathcal{L}_0$ implies the following lemma.

Lemma 2.7. *Let $\langle T, Ab, \overline{}\rangle$ be the assumption-based framework corresponding to a default theory (T, D). Then for each assumption $M\alpha \in Ab$ and for each set of assumptions $\Delta \subseteq Ab$*

$$T \cup \Delta \vdash M\alpha \quad \textit{if and only if} \quad M\alpha \in \Delta.$$

This lemma is important because, in general, frameworks which satisfy the property

$$\forall\beta \in Ab, \quad T \cup \Delta \vdash \beta \text{ if and only if } \beta \in \Delta,$$

(called "flatness" in Definition 4.10) are guaranteed to have sensible semantics, as we will see in Corollary 4.11 and Theorem 6.2.

Example 2.8. There are several ways of expressing the "innocent-unless-guilty" example in default logic. Assume that the vocabulary of the language $\mathcal{L}_0$ consists of the propositional symbols *innocent* and *guilty*.

(1) Similarly to Example 2.5 of Theorist, the default theory is

$$T = \{\neg guilty \rightarrow innocent\},$$
$$D = \left\{\frac{M\neg guilty}{\neg guilty}\right\}.$$

In the corresponding assumption-based framework there is only one maximal conflict-free set of assumptions Δ containing $M\neg guilty$.

[8] In Reiter's original formulation, default rules are expressed in the slightly different form $\frac{\alpha : M\beta_1, \ldots, M\beta_n}{\gamma}$

(2) The default theory is

$$T=\emptyset,$$
$$D=\left\{\frac{M\neg guilty}{innocent}\right\}.$$

The corresponding assumption-based framework has one maximal conflict-free set of assumptions as in representation 1.

The default theory in the first part of this example is a *normal default theory* [49], i.e. with all default rules of the form

$$\frac{M\alpha}{\alpha}$$

Poole [43] shows that there is a one-to-one correspondence between normal default theories (T, D) and abductive frameworks (T, Ab) in Theorist, where each normal default $\frac{M\alpha}{\alpha}$ in D corresponds to an assumption α in Ab and vice versa. Moreover, under this correspondence, the semantics [49] of normal default theories coincides with the naive semantics of Theorist. These results also follow from more general results we will present later, in Section 3.3.

Marek, Nerode and Remmel [36,37] generalise default logic by dropping the condition that the underlining monotonic logic be classical first-order logic. The resulting *non-monotonic rule system* is defined to be a pair $(\mathcal{L}_0, \mathcal{R})$ where $\mathcal{L}_0$ is a nonempty set of sentences and $\mathcal{R}$ is a set of non-monotonic rules of the form [9]

$$\frac{\alpha_1, \ldots, \alpha_n, M\beta_1, \ldots, M\beta_m}{\gamma}$$

where $\alpha_1, \ldots, \alpha_n, \beta_1, \ldots, \beta_m, \gamma \in \mathcal{L}_0$. If $m = 0$ then the rule is an inference rule of some "underlying" monotonic logic. Otherwise, the rule is similar to a default rule of default logic.

A theory $T \subseteq \mathcal{L}_0$ with respect to a non-monotonic rule system $(\mathcal{L}_0, \mathcal{R})$ can be viewed as an assumption-based framework $\langle T, Ab, \overline{}\rangle$ with respect to a deductive system $(\mathcal{L}, \mathcal{R})$ where

- $\mathcal{L} = \mathcal{L}_0 \cup \{M\alpha \mid \alpha \in \mathcal{L}_0\}$,
- $T \subseteq \mathcal{L}_0$,
- $Ab = \{M\alpha \mid \alpha \in \mathcal{L}_0\}$, and
- $\overline{M\alpha} = \neg\alpha$.

2.4. Autoepistemic logic

Let $(\mathcal{L}, \mathcal{R})$ be a deductive system where $\mathcal{L}$ is a modal language containing a modal operator L, and $\mathcal{R}$ is some set of inference rules for classical logic for the language $\mathcal{L}$. The intended meaning of $L\alpha$ is that α is believed.

[9] Marek, Nerode and Remmel's original notation is $\frac{\alpha_1, \ldots, \alpha_n : \beta_1, \ldots, \beta_m}{\gamma}$

A theory T in *autoepistemic logic* [40] can be viewed as an assumption-based framework $\langle T, Ab, \overline{\ } \rangle$, where

- $Ab = \{L\alpha \mid \alpha \in \mathcal{L}\} \cup \{\neg L\alpha \mid \alpha \in \mathcal{L}\}$
- $\overline{\neg L\alpha} = \alpha$ and $\overline{L\alpha} = \neg L\alpha$ for each $\alpha \in \mathcal{L}$.

Example 2.9. The "innocent-unless-guilty" example can be expressed naturally in autoepistemic logic by the following theory

$$\{\neg Lguilty \rightarrow innocent\}$$

In the corresponding assumption-based framework there are two maximal conflict-free extensions. One contains the assumption $\neg Lguilty$, the other contains the assumption $\neg Linnocent$. Only extensions of the first kind are acceptable in the standard semantics of autoepistemic logic, which we will investigate in Section 3.4.

2.5. Non-monotonic modal logics

Non-monotonic modal logics [39] can be formulated in terms of deductive systems of the form $(\mathcal{L}, \mathcal{R})$ where $\mathcal{L}$ is a first-order modal language containing a modal operator, L, and $\mathcal{R}$ is some set of inference rules for the language $\mathcal{L}$. Different choices for $\mathcal{R}$ correspond to different modal logics. However, all $\mathcal{R}$ contain all instances of the *necessitation rule*: [10]

$$\frac{\alpha}{L\alpha} \quad \text{for all } \alpha \in \mathcal{L}.$$

Given a theory $T \subseteq \mathcal{L}$, the corresponding assumption-based framework is $\langle T, Ab, \overline{\ } \rangle$ where

- $Ab = \{\neg L\alpha \mid \alpha \in \mathcal{L}\}$;
- $\overline{\neg L\alpha} = \alpha$ for each $\alpha \in \mathcal{L}$.

Example 2.10. Let T be the theory

$$\{\neg Lguilty \rightarrow innocent\}.$$

This has the same two kinds of maximal conflict-free extensions as in Example 2.9 in autoepistemic logic, but containing only negative assumptions. Similarly, only the first kind of extension, containing the assumption $\neg Lguilty$, is acceptable in the standard semantics, which is an instance of the stable semantics defined in the next section.

In this example, the naive semantics for autoepistemic and non-monotonic modal logic coincide. More generally, for some choices of $\mathcal{R}$, autoepistemic and non-monotonic modal logics coincide (e.g. see [53]), where for others they differ.

[10] Here we consider the necessitation rule as formulated in [39]. However, note that in monotonic modal logics necessitation is restricted to sentences $\alpha \in \mathcal{L}$ that are first-order tautologies.

3. Stable semantics

In this section we define the notion of stable semantics, which corresponds to most of the credulous semantics which have been proposed for default reasoning, including Theorist's extensions [43], the stable model semantics of logic programming [20], extensions in default logic [49], expansions in autoepistemic logic [40] and fixed points in non-monotonic modal logics [39].

Informally, a set of assumptions is stable if it is conflict-free and it attacks (by proving the contrary of) every assumption it does not contain. More formally

Definition 3.1. Given an assumption-based framework $\langle T, Ab, \bar{\ } \rangle$,

- a set of assumptions $\Delta \subseteq Ab$ *attacks* an assumption $\alpha \in Ab$ if and only if $T \cup \Delta \vdash \overline{\alpha}$,
- a set of assumptions $\Delta \subseteq Ab$ *attacks* a set of assumptions $\Delta' \subseteq Ab$ if and only if Δ attacks some assumption $\alpha \in \Delta'$.

If Δ attacks α (respectively Δ') we also say that Δ is an *attack against* α (respectively Δ'). Notice that an immediate consequence of Definition 3.1 is that

- given a set of assumptions $\Delta \subseteq Ab$, if Δ is conflict-free then Δ does not attack itself.

However, the converse implication does not hold in general, because Δ might attack an assumption which is implied by $T \cup \Delta$ but is not in Δ explicitly, as illustrated by the following example.

Example 3.2. Consider the autoepistemic logic theory T

$$\{\neg Lp \rightarrow \neg Lq,\ q\}$$

and the set of assumptions $\Delta = \{\neg Lp\}$. Δ does not attack itself, since $T \cup \Delta \not\vdash p$. However, Δ is not conflict-free, since $T \cup \Delta \vdash \neg Lq, q$.

If a set of assumptions does not attack itself and explicitly contains all the assumptions which, together with the given theory, it implies, then it is conflict-free. More formally:

Definition 3.3. Given an assumption-based framework $\langle T, Ab, \bar{\ } \rangle$, a set of assumptions $\Delta \subseteq Ab$ is *closed* if and only if $\Delta = \{\alpha \in Ab \mid T \cup \Delta \vdash \alpha\}$.

It follows immediately that

- a closed set of assumptions $\Delta \subseteq Ab$ is conflict-free if and only if Δ does not attack itself.

Note that a maximal conflict-free set of assumptions is necessarily closed.

Assumption-based frameworks where all sets of assumptions are closed are simpler than other frameworks. In Section 4, such special frameworks are said to be flat (see Definition 4.10).

Definition 3.4. A set of assumptions Δ is *stable* if and only if

- Δ is closed
- Δ does not attack itself and
- Δ attacks each assumption $\alpha \notin \Delta$.

The stable semantics generalises the naive semantics, as shown by the following theorem.

Theorem 3.5. *For any assumption-based framework $\langle T, Ab, \overline{\ } \rangle$, for any set of assumptions $\Delta \subseteq Ab$,*

if Δ is stable then Δ is maximal conflict-free.

Proof. Assume Δ is stable. Then Δ is conflict-free. Therefore, we need to show only that, for each assumption $\alpha \notin \Delta$, $\Delta \cup \{\alpha\}$ is not conflict-free. But this follows directly from the fact that for each assumption $\alpha \notin \Delta$, Δ attacks α. □

The converse of Theorem 3.5 does not hold in general, as illustrated by the logic programming formulation of the "innocent-unless-guilty" example in Example 2.6. Here, the only stable set of assumptions is $\Delta_1 = \{not\ guilty\}$. In fact, the (maximal) conflict-free set of assumptions $\Delta_2 = \{not\ innocent\}$ does not attack *not guilty*.

The assumption-based frameworks for which the stable semantics and the naive semantics coincide are called *normal* assumption-based frameworks.

Definition 3.6. An assumption-based framework $\langle T, Ab, \overline{\ } \rangle$ is *normal* if and only if every maximal conflict-free set of assumptions is stable.

The following theorem gives a sufficient condition for assumption-based framework to be normal.

Theorem 3.7. *An assumption-based framework $\langle T, Ab, \overline{\ } \rangle$ is normal if for each $\Delta \subseteq Ab$ and each assumption $\alpha \notin \Delta$*

if $\Delta \cup \{\alpha\}$ is not conflict-free then Δ attacks α.

Proof. From Theorem 3.5, if Δ is stable then Δ is maximal conflict-free. Suppose Δ is maximal conflict-free. Then, it is closed and does not attack itself. Since for each assumption $\alpha \notin \Delta$, $\Delta \cup \{\alpha\}$ is not conflict free. Hence Δ attacks α. It is obvious that Δ is stable. □

The following theorem provides an alternative characterisation of stability.

Definition 3.8. Given an assumption-based framework $\langle T, Ab, \overline{\ } \rangle$ and a set of assumptions $\Delta \subseteq Ab$,

$$\mathcal{S}(\Delta) = \{\alpha \mid \Delta \text{ does not attack } \alpha\}.$$

Theorem 3.9. *A closed set of assumptions Δ is stable if and only if $\Delta = \mathcal{S}(\Delta)$.*

Proof. Let Δ be a closed set of assumptions. Then

- Δ does not attack itself if and only if $\Delta \subseteq \mathcal{S}(\Delta)$;
- Δ attacks each assumption $\alpha \notin \Delta$ if and only if $\mathcal{S}(\Delta) \subseteq \Delta$. □

The notion of *stable extension*, i.e. of a theory $Th(T \cup \Delta)$ for some stable set of assumptions Δ, corresponds, more closely than the notion of stable set of assumptions, to the standard semantics of most non-monotonic logics, as we will see later in this section. Note that the set of assumptions contained in a stable extension is automatically closed.

The following theorem provides four alternative characterisations of the notion of stable extension. The theorem shows that the different characterisations differ primarily in the way they characterise theoremhood in the underlying monotonic logic. The first two characterisations are the simplest, because they take the notion of monotonic theoremhood to be already given. The second, in particular, corresponds to the standard characterisation of stable models in logic programming, extensions in autoepistemic logic and fixed points in non-monotonic modal logics. The third characterises a sentence as a theorem if it is derivable by means of a finite number of inference steps. The fourth characterises the set of all theorems as the smallest set containing an initial theory $(T \cup \Delta_E)$ and closed under the operation of adding theorems. The fourth corresponds to the original definition of extension in default logic given in [49], whereas the third corresponds to the equivalent characterisation of default logic as proved in [49].

Theorem 3.10. *Given an assumption-based framework $\langle T, Ab, \overline{} \rangle$ with respect to $(\mathcal{L}, \mathcal{R})$ and $E \subseteq \mathcal{L}$, let $\Delta_E = \{\alpha \in Ab \mid \overline{\alpha} \notin E\}$. Then the following statements are equivalent:*

(1) *E is a stable extension of $\langle T, Ab, \overline{} \rangle$.*

(2) *$E = Th(T \cup \Delta_E)$, and Δ_E is closed.*

(3) *$E = \bigcup_{i=1}^{\infty} E_i$, where*
- *$E_1 = T \cup \Delta_E$,*
- *for each $i > 1$ $E_{i+1} = E_i \cup \{\beta \in \mathcal{L} \mid \frac{\alpha_1, \ldots, \alpha_n}{\beta} \in \mathcal{R}$ and $\alpha_1, \ldots, \alpha_n \in E_i\}$,*

and Δ_E is closed.

(4) *$E = \Gamma(E)$ where for each set $S \subseteq \mathcal{L}$, $\Gamma(S)$ is the smallest set such that*
- *$T \cup \Delta_S \subseteq \Gamma(S)$, where $\Delta_S = \{\alpha \in Ab \mid \overline{\alpha} \notin S\}$,*
- *for each $\frac{\alpha_1, \ldots, \alpha_n}{\beta} \in \mathcal{R}$, if $\alpha_1, \ldots, \alpha_n \in \Gamma(S)$ then $\beta \in \Gamma(S)$,*

and Δ_E is closed.

Proof. (1) $\Leftrightarrow$ (2)

E is a stable extension

(by definition)

there exists Δ such that $E = Th(T \cup \Delta)$ and Δ is stable

(by Theorem 3.9)

$E = Th(T \cup \Delta)$ where $\Delta = \{\alpha \in Ab \mid \Delta$ does not attack $\alpha\}$ and Δ is closed

(by definition of attack)

$E = Th(T \cup \Delta)$ where $\Delta = \{\alpha \in Ab \mid T \cup \Delta \not\vdash \overline{\alpha}\}$ and Δ is closed

(by definition of E)

$E = Th(T \cup \Delta)$ where $\Delta = \{\alpha \in Ab \mid \overline{\alpha} \notin E\}$ and Δ is closed

(by definition of Δ_E)

$E = Th(T \cup \Delta_E)$ and Δ_E is closed.

(2) $\Leftrightarrow$ (3) By definition of *Th*, E_i is the set of theorems derivable from the theory $T \cup \Delta_E$ by means of a deduction of length i.

(2) $\Leftrightarrow$ (4) $\Gamma(S)$ is the smallest set containing $T \cup \Delta_S$ and closed under theoremhood. Therefore $\Gamma(S) = Th(T \cup \Delta_S)$ and the condition $E = \Gamma(E)$ is equivalent to $E = Th(T \cup \Delta_E)$ in (2). □

3.1. Theorist

Theorem 3.11. *For any abductive framework (T, Ab), the corresponding assumption-based framework $\langle T, Ab, \overline{} \rangle$ is normal.*

Proof. Suppose there exist $\Delta \subseteq Ab$ and $\alpha \in Ab$, $\alpha \notin \Delta$ such that $\Delta \cup \{\alpha\}$ is not conflict-free. Then, because an inconsistency in classical logic implies any sentence, $T \cup \Delta \cup \{\alpha\} \vdash \neg\alpha$. Then $T \cup \Delta \vdash \alpha \rightarrow \neg\alpha$, and therefore $T \cup \Delta \vdash \neg\alpha$, i.e. Δ attacks α. Therefore the normality of the considered assumption-based framework follows immediately from Theorem 3.7. □

It follows directly from this theorem, from Theorems 3.5 and 2.4 and from Definition 3.6

Theorem 3.12. *Given a Theorist abductive framework (T, Ab), E is an extension of (T, Ab) in the sense of* [43] *if and only if E is a stable extension of the corresponding assumption-based framework.*

3.2. Logic programming

Given a normal logic program P, let $\langle P, Ab, \overline{} \rangle$ be the corresponding assumption-based framework (as defined in Section 2.2).

By Theorem 3.10, equivalence between (1) and (2), E is a stable extension if and only if $E = \{q \mid P \cup \Delta_E \vdash q\}$ where $\Delta_E = \{not\, p \in Ab \mid p \notin E\}$. Note that the condition that Δ_E is closed is unnecessary, because every set of assumptions in such an assumption-based framework is closed.

Theorem 3.13 below states that stable semantics for logic programming corresponds to stable model semantics [20], defined in terms of Herbrand models.

A *Herbrand interpretation* I of a theory is any subset of the Herbrand base of the language of the theory. It assigns the truth value *true* to any ground atom in I and the truth value *false* to any ground atom not in I. The truth value of any other sentence is defined in the usual way. A *Herbrand model* of a theory is a Herbrand interpretation in which every sentence in the theory is *true*.

By definition [20], M is a *stable model* of P if and only if M is the least Herbrand model of the program P_M obtained by eliminating from P:

- all clauses with conditions of the form *not* p such that $p \in M$,
- all negative literals from the remaining clauses.

It is easy to see that the least Herbrand model of P_M coincides with the set $\{p \in \mathcal{HB} \mid P \cup \Delta_M \vdash p\}$ where $\Delta_M = \{not\ p \mid p \notin M\}$. Therefore, M is a stable model of P if and only if $M = \{p \in \mathcal{HB} \mid P \cup \Delta_M \vdash p\}$. As a consequence, the following theorem holds:

Theorem 3.13. *M is a stable model in the sense of* [20] *of a logic program P if and only if there is a stable extension E of the corresponding assumption-based framework such that $M = E \cap \mathcal{HB}$.*

It is similarly easy to show that there is a one-to-one correspondence between answer sets [21] of extended logic programs and stable extensions.

Notice that the notion of stable model (and similarly of answer set) is purely syntactic. Extensions E are turned into models simply by restricting attention to the variable-free atoms or literals in E. This close correspondence between extensions and models suggests that there is no strong reason to prefer a model theoretic semantics over a purely syntactic one based on extensions. In fact, for our purposes, it is more convenient to deal with sets of assumptions than with extensions or models. This will become more apparent when we investigate the admissibility semantics in the next section.

3.3. Default logic

Given a deductive system $(\mathcal{L}_0, \mathcal{R}_0)$ for first-order logic and a default theory (T, D), let $\langle T, Ab, \overline{\ }\rangle$ be the corresponding assumption-based framework with respect to $(\mathcal{L}, \mathcal{R}_0 \cup D)$.

Reiter [49] defines a set $E \subseteq \mathcal{L}_0$ to be an *extension* of (T, D) if and only if $E = \Gamma_0(E)$ where Γ_0 is defined as follows: for each set $S \subseteq \mathcal{L}_0$, $\Gamma_0(S)$ is the smallest set such that

- $T \subseteq \Gamma_0(S)$,
- $\Gamma_0(S)$ is closed with respect to the first order deductive system $(\mathcal{L}_0, \mathcal{R}_0)$, and
- for each $\frac{\alpha, M\beta_1, \ldots, M\beta_n}{\gamma} \in D$ if $\alpha \in \Gamma_0(S)$ and $\neg\beta_i \notin S$ for each $1 \leqslant i \leqslant n$ then $\gamma \in \Gamma_0(S)$

From the definition of $\Delta_S = \{M\beta \mid \neg\beta \notin S\}$ (given in Theorem 3.10), it follows immediately that

Lemma 3.14. *Let $S \subseteq \mathcal{L}_0$. Then $\Gamma_0(S)$ is the smallest set such that*

- *$T \subseteq \Gamma_0(S)$,*
- *for each $\frac{\alpha_1, \ldots, \alpha_n}{\gamma} \in \mathcal{R}_0 \cup D$ if $\Gamma_0(S) \cup \Delta_S \vdash \alpha_i$ for each $1 \leqslant i \leqslant n$ then $\gamma \in \Gamma_0(S)$.*

For any $S' \subseteq \mathcal{L}$, let $\Gamma(S') = \Gamma_0(S' \cap \mathcal{L}_0) \cup \Delta_{S'}$.

From the flatness of default theories (Theorem 4.12), it follows directly that

Lemma 3.15. *$\Gamma(S')$ is the smallest set such that*

- $T \cup \Delta_{S'} \subseteq \Gamma(S')$,
- *$\Gamma(S')$ is closed with respect to the deductive system $(\mathcal{L}, \mathcal{R})$.*

Now it follows directly from Theorem 3.10, equivalence between (1) and (4), that

Theorem 3.16. *$E \subseteq \mathcal{L}_0$ is an extension in the sense of* [49] *of a default theory (T, D) if and only if there is a stable extension E' of the corresponding assumption-based framework such that $E = E' \cap \mathcal{L}_0$.*

A similar result holds for non-monotonic rule systems [36,37]. Namely E is an extension of a theory T in a non-monotonic rule system $(\mathcal{L}, \mathcal{R})$ if and only if there is a stable extension E' of the corresponding assumption-based framework such that $E = E' \cap \mathcal{L}$.

This result follows directly from Theorem 3.10, equivalence between (1) and (3).

The assumption-based frameworks corresponding to normal default theories are normal in the sense of Definition 3.6:

Theorem 3.17. *For any normal default theory (T, D), the corresponding assumption-based framework $\langle T, Ab, \overline{\ } \rangle$ is normal.*

Proof. The proof is similar to that of Theorem 3.11. Suppose there exist $\Delta \subseteq Ab$ and $M\alpha \in Ab$, $M\alpha \notin \Delta$, such that $\Delta \cup \{M\alpha\}$ is not conflict-free. Then, there exists $M\beta \in \Delta \cup \{M\alpha\}$ such that $T \cup \Delta \cup \{M\alpha\} \vdash \neg\beta$. But $M\beta$ occurs in (T, D) only in a default rule $\frac{M\beta}{\beta} \in D$. Therefore, $T \cup \Delta \cup \{M\alpha\} \vdash \beta$, and therefore $T \cup \Delta \cup \{M\alpha\}$ is inconsistent and implies every sentence in $\mathcal{L}_0$. In particular, $T \cup \Delta \cup \{M\alpha\} \vdash \neg\alpha$. But, as before, $M\alpha$ occurs in (T, D) only in a default rule $\frac{M\alpha}{\alpha} \in D$. Therefore, $T \cup \Delta \cup \{\alpha\} \vdash \neg\alpha$, and therefore $T \cup \Delta \vdash \neg\alpha$, i.e. Δ attacks $M\alpha$. Hence, following Theorem 3.7, $\langle T, Ab, \overline{\ } \rangle$ is normal. □

3.4. Autoepistemic logic

Given a modal language $(\mathcal{L}, \mathcal{R})$ containing a modal operator L and an autoepistemic theory $T \subseteq \mathcal{L}$, let $\langle T, Ab, \overline{\ } \rangle$ be the corresponding assumption-based framework.

By Theorem 3.10, equivalence between (1) and (2), E is a stable extension if and only if $E = Th(T \cup \Delta_E)$ where $\Delta_E = \{L\alpha \in Ab \mid \neg L\alpha \notin E\} \cup \{\neg L\alpha \in Ab \mid \alpha \notin E\}$ and Δ_E is closed.

The following theorem shows the correspondence between stable extensions and the original stable expansion semantics of autoepistemic logic given in [40]: E is a *stable expansion* of T in the sense of [40] if and only if $E = Th(T \cup \{L\alpha \mid \alpha \in E\} \cup \{\neg L\alpha \mid \alpha \notin E\})$.

In the proof of the theorem we will refer to the fact that a consistent theory T can admit an inconsistent stable expansion. For example, $T = \{\neg Lp\}$ has the stable expansion $E = \{Lp, \neg Lp, \ldots\} = \mathcal{L}$.

Theorem 3.18. *A theory E is a stable extension of the assumption-based framework corresponding to an autoepistemic theory T if and only if E is consistent and is a stable expansion* [40] *of T.*

Proof. ($\Leftarrow$) Assume that E is a stable expansion and E is consistent. We need to prove only that (1) $\{L\alpha \in Ab \mid \neg L\alpha \notin E\} = \{L\alpha \mid \alpha \in E\}$ and (2) $\Delta = \{L\alpha \mid \alpha \in E\} \cup \{\neg L\alpha \mid \alpha \notin E\}$ is closed:

(1) But $\neg L\alpha \notin E$ implies (since, by definition of stable expansion, $\alpha \notin E$ implies $\neg L\alpha \in E$) $\alpha \in E$. Conversely $\alpha \in E$ implies (by definition of stable expansion) $L\alpha \in E$, that in turn implies (because E is consistent) $\neg L\alpha \notin E$.

(2) Assume that Δ is not closed. Then, either there exists $L\alpha \in E$ such that $L\alpha \notin \Delta$ or there exists $\neg L\alpha \in E$ such that $\neg L\alpha \notin \Delta$. In the first case, if $L\alpha \notin \Delta$, then $\alpha \notin E$, then $\neg L\alpha \in E$ and E is inconsistent. In the second case, if $\neg L\alpha \notin \Delta$, then $\alpha \in E$, then $L\alpha \in E$ and E is inconsistent. Therefore Δ is closed.

($\Rightarrow$) Assume that E is a stable extension of the assumption-based framework corresponding to T. We need to prove only that (1) E is consistent, and (2) $\{L\alpha \in Ab \mid \neg L\alpha \notin E\} = \{L\alpha \mid \alpha \in E\}$, i.e. $\neg L\alpha \notin E$ if and only if $\alpha \in E$:

(1) Otherwise E would not be a conflict-free extension and therefore would not be stable.

(2) But $\neg L\alpha \notin E$, if and only if (since $E = Th(T \cup \Delta_E)$ and Δ_E is closed) $\neg L\alpha \notin \Delta_E$, if and only if (by definition of Δ_E) $\alpha \in E$. □

3.5. Non-monotonic modal logics

Given a first-order modal language $(\mathcal{L}, \mathcal{R})$ containing a modal operator L and a non-monotonic modal theory $T \subseteq \mathcal{L}$, let $\langle T, Ab, \overline{\ \ } \rangle$ be the corresponding assumption-based framework.

By Theorem 3.10, E is a stable extension if and only if $E = Th(T \cup \Delta_E)$ where $\Delta_E = \{\neg L\alpha \in Ab \mid \alpha \notin E\}$ and Δ_E is closed.

The following theorem shows the correspondence between stable extensions and the original fixed point semantics for non-monotonic modal logics given in [39]: E is a *fixed point* of T if and only if $E = Th(T \cup \{\neg L\alpha \mid \alpha \notin E\})$.

In the proof of the theorem we will use the property, following directly from the definition of fixed point, that a fixed point E of a theory T is inconsistent only if T is inconsistent. Therefore, differently from the case of autoepistemic logic, it is sufficient to assume that the theory T is consistent to guarantee the correspondence between stable extensions and fixed points.

Theorem 3.19. *A theory E is a stable extension of the assumption-based framework corresponding to a non-monotonic modal theory T if and only if E is a fixed point of T and T is consistent.*

Proof. ($\Rightarrow$) We need to prove only that T is consistent. But if T was inconsistent then $E = \mathcal{L}$ would not be a stable extension, since $\Delta_E = \emptyset$ would not be closed.

($\Leftarrow$) Assume that E is a fixed point of T and that T is consistent. We need to prove only that $\Delta_E = \{\neg L\alpha \in Ab \mid \alpha \notin E\}$ is closed. Suppose that it is not. Then, there exists $\neg L\alpha \in E$ such that $\neg L\alpha \notin \Delta_E$. But then, by definition of fixed point, if $\neg L\alpha \notin \Delta_E$ then $\alpha \in E$. By necessitation, $L\alpha \in E$. Therefore, E is inconsistent. This contradicts the hypothesis that T is consistent. □

4. Admissibility semantics

Viewed from an argumentation-theoretic point of view, stable semantics seems unnecessarily restrictive, because it insists that a set of assumptions should take a stand on every issue. On the other hand, the naive semantics, which allows any conflict-free extension, is too liberal, because it allows intuitively unacceptable sets of assumptions. We need a semantics which is more tolerant than stable semantics and less liberal than naive semantics. Such a semantics, called the admissibility semantics, was introduced for logic programming by Dung [10]. It provides a semantics in cases like those in Examples 4.1 and 4.2, where a stable semantics does not exist.

Example 4.1. Consider the logic program

$$\{p \leftarrow \mathit{not}\ p\}.$$

This has no stable extensions. However, $\Delta = \emptyset$ is admissible in the intuitive sense that Δ is conflict-free and it is not attacked by any other set of assumptions. Moreover, Δ is maximal admissible, because the only larger set $\{\mathit{not}\ p\}$ attacks itself.

Example 4.2. Consider the autoepistemic and non-monotonic modal theory

$$\{\neg Ls \rightarrow \neg r,\ \ \neg Lt \rightarrow r\}.$$

This has no stable extension. In fact, if it had a stable extension $E = Th(T \cup \Delta)$, with Δ a stable set of assumptions, then either Δ would contain $\neg Ls$ and $\neg Lr$ or not. In the first case, Δ would attack itself and therefore would not be stable. In the second case, Δ would be unable to attack all assumptions not in Δ. However, both $\Delta_1 = \{\neg Ls\}$ and $\Delta_2 = \{\neg Lt\}$ are admissible, because each is conflict-free and can defend itself against any closed attack. In particular, any attack against Δ_1 or Δ_2 must contain the inconsistent set $\{\neg Ls, \neg Lt\}$. Any closed attack, therefore, contains both $\neg L\neg r$ and $\neg Lr$, one of which is attacked by Δ_1 or Δ_2.

Definition 4.3. A closed set of assumptions $\Delta \subseteq Ab$ is *admissible* if and only if

- Δ does not attack itself, and
- for each closed set of assumptions $\Delta' \subseteq Ab$, if Δ' attacks Δ then Δ attacks Δ'.

It is easy to see that in any assumption-based framework whose underlining deductive system contains a notion of inconsistency such that inconsistency implies everything, admissible sets of assumptions are consistent.

Definition 4.4. A set of assumptions $\Delta \subseteq Ab$ is *preferred* if and only if Δ is maximal (with respect to set inclusion) admissible.

As mentioned above, the notions of admissible and preferred sets of assumptions generalise the semantics for logic programming given by Dung [10]. This is expressed by the following theorem.

Theorem 4.5. *For each logic program T and set of assumptions Δ in the assumption-based framework $\langle T, Ab, \overline{\ \ } \rangle$ corresponding to T, $T \cup \Delta$ is an admissible scenario of T ($T \cup \Delta$ is a preferred extension of T) in the sense of* [10] *if and only if Δ is an admissible (preferred) set of assumptions in $\langle T, Ab, \overline{\ \ } \rangle$.*

This theorem follows directly from the characterisation of Dung's admissible scenarios and preferred extensions given in [25,26].

Throughout this section, we focus our attention on admissible sets of assumptions rather than on admissible and preferred extensions. However, the restriction that admissible sets Δ be closed means that they are like extensions in the sense that, whereas extensions contain all the sentences $Th(T \cup \Delta)$ derivable from $T \cup \Delta$, closed sets of assumptions contain all the assumptions $Th(T \cup \Delta) \cap Ab$ derivable.

Instead of understanding semantics in terms of admissible extensions or sets of assumptions, it is also possible to define semantics in terms of the ground literals in $E = Th(T \cup \Delta)$. In the case of logic programming, by assigning *true* to a ground atom p if $p \in E$ and *false* to a ground atom p if *not* $p \in E$, we obtain a three-valued model of T. It follows directly from the result shown in [29], that there is a one-to-one correspondence between partial stable models [50] and models corresponding to preferred sets of assumptions.

It is also easy to show that there is a one-to-one correspondence between admissible and preferred sets of assumptions and the semantics of extended logic programs proposed by Dung and Ruamviboonsuk [14].

The following theorem shows that preferred sets of assumptions provide a strictly more liberal semantics than stable sets of assumptions.

Theorem 4.6. *Every stable set of assumptions is preferred but not every preferred set is stable.*

Proof. Let Δ be a stable set of assumptions. First we show that Δ is admissible. Let Δ' be an arbitrary (closed) attack against Δ. Since Δ does not attack itself, it is clear that $\Delta' \not\subseteq \Delta$. Hence, $\Delta' - \Delta$ is not empty. Since Δ is stable, Δ attacks $\Delta' - \Delta$. Therefore Δ attacks Δ'. So Δ is admissible. Since Δ attacks every assumption not belonging to it, it is clear that Δ is a maximal admissible set of assumptions. Hence Δ is preferred.

Example 4.1 shows that not every preferred set of assumptions is stable. □

In general, maximal conflict-free sets of assumptions need not be preferred, as shown by Example 2.6, where the only preferred set of assumptions is $\{not\ guilty\}$. Moreover, preferred sets of assumptions are not necessarily maximal conflict-free, as shown by the following example.

Example 4.7. In the assumption-based framework corresponding to the logic program

$$\{p \leftarrow not\ q,\ \ q \leftarrow not\ r,\ \ r \leftarrow not\ p\}$$

there is only one preferred set of assumptions, namely $\emptyset$, which is not maximal conflict-free. In fact, the maximal conflict-free sets of assumptions are $\{not\ p\}$, $\{not\ q\}$ and $\{not\ r\}$, which are not admissible.

However, the naive, stable and preferred semantics coincide for normal assumption-based frameworks, as stated in the following theorem:

Theorem 4.8. *For any normal assumption-based framework $\langle T, Ab, \overline{\ }\rangle$, for any set of assumptions $\Delta \subseteq Ab$, the following statements are equivalent:*
(1) *Δ is maximal conflict-free.*
(2) *Δ is stable.*
(3) *Δ is preferred.*

Proof. (1) ⇒ (2) By Definition 3.6 of normal assumption-based framework.

(2) ⇒ (3) By Theorem 4.6.

(3) ⇒ (1) Suppose Δ is preferred, but not maximal conflict-free. Then, Δ is conflict-free because it is preferred. Therefore, by Theorem 2.3, there exists $\Delta' \supset \Delta$ such that Δ' is maximal conflict-free. Since $\langle T, Ab, \overline{\ }\rangle$ is normal, Δ' is stable. By Theorem 4.6, Δ' is preferred, thus contradicting the hypothesis that Δ is preferred. □

The following theorem and its corollary guarantee the existence of preferred sets of assumptions.

Theorem 4.9. *For every admissible set of assumptions Δ, there exists a preferred set of assumptions which contains Δ.*

Proof. The set of all admissible sets of assumptions that are supersets of Δ is a non-empty partial order with respect to subset inclusion. Let $\Delta_0, \Delta_1, \ldots, \Delta_n, \ldots$, where n is an ordinal number, be any increasing sequence of admissible sets of assumptions such that $\Delta_0 = \Delta$. It is easy to see that this sequence has an upper bound $\Delta' = \bigcup_{i \geqslant 0} \Delta_i$ which is also admissible: if Δ' attacked itself then some finite subset of Δ', contained in some Δ_i, would attack itself, thus contradicting the admissibility of Δ_i. Similarly, any attack against Δ' is an attack against some Δ_i. The admissibility of Δ_i implies that Δ_i and therefore Δ' counter attacks this attack. Therefore, by Zorn's lemma, since every increasing sequence of admissible sets that are supersets of Δ has an upper bound, then there exists a maximal admissible set of assumptions containing Δ. □

It follows directly from this theorem that, if at least one admissible set of assumptions exists, then there also exists a preferred set. It is easy to see that if the empty set of assumptions is closed, then it is also admissible. This property holds trivially for flat frameworks, defined as follows:

Definition 4.10. An assumption-based framework is said to be *flat* if and only if every set of assumptions $\Delta \subseteq Ab$ is closed.

Corollary 4.11. *Every flat assumption-based framework possesses at least one preferred extension.*

Flat assumption-based frameworks have a flat structure, in the sense that all assumptions are independent from one other. In general, in a non-flat assumption-based framework, an assumption α can be implied by a set of assumptions Δ for one of two reasons:

- Δ is inconsistent with the theory, and in the underlying monotonic logic inconsistency implies any sentence, including α.
- α can be derived from Δ by means of the domain-specific theory, T.

Implicit assumptions of the first kind can arise in Theorist, autoepistemic logic and non-monotonic modal logics and, as we will see later, in Section 6.2, in circumscription. Implicit assumptions of the second kind can arise in Theorist, circumscription, autoepistemic logic and non-monotonic modal logics, as illustrated in Example 3.2. However, it is easy to see that neither kind of implicit assumption can arise in logic programming and in our formulation of default logic (see Lemma 2.7 for default logic). Therefore:

Theorem 4.12.

- *The assumption-based framework corresponding to any logic program is flat.*
- *The assumption-based framework corresponding to any default theory is flat.*

However, the assumption-based frameworks corresponding to autoepistemic theories are never flat, since the set of assumptions $\{L\alpha, \neg L\alpha\}$, for any sentence α, is inconsistent for any theory T. The assumption-based frameworks corresponding to Theorist or non-monotonic modal theories may be flat in some cases, but are not flat in general. For example, the assumption-based framework corresponding to the non-monotonic modal theory

$$\{p\}$$

is flat, while the assumption-based framework corresponding to

$$\{\neg Lp\}$$

is not.

Although, arguably, it is an improvement over both the naive and the stable semantics, admissibility semantics can itself be improved, as the following example shows.

Example 4.13. Consider the logic program P

$$\{r^* \leftarrow \textit{not } s,\ \ r \leftarrow \textit{not } t,\ \ s \leftarrow r, r^*,\ \ t \leftarrow r, r^*\}$$

which simulates, in part, the autoepistemic and non-monotonic modal theory of Example 4.2. The positive atom r^* simulates the negative literal $\neg r$; and the last two clauses partially simulate the property in classical logic that an inconsistency implies anything. P also partly simulates the extended logic program

$\{\sim r \leftarrow \mathit{not}\ s,\ \ r \leftarrow \mathit{not}\ t\}.$

Like the theory T of Example 4.2, P has no stable extensions. However, unlike T, the sets $\Delta_1 = \{\mathit{not}\ s\}$ and $\Delta_2 = \{\mathit{not}\ t\}$ are not admissible, because the closed attack $\Delta' = \{\mathit{not}\ s, \mathit{not}\ t\}$, against both Δ_1 and Δ_2, cannot be counterattacked by Δ_1 and Δ_2. Intuitively, however, Δ_1 and Δ_2 are both "acceptable" because Δ' attacks itself and is therefore not an "acceptable" attack.

Two semantics, called "stable theory" and "acceptability" semantics, have been proposed for logic programming by Kakas and Mancarella [28], to deal with cases like the one in this example. These semantics can be generalised and defined more abstractly for any assumption-based framework. These generalisations are straight-forward, and we shall not discuss them further in this paper. A formal definition of these generalisations can be found in [27].

5. Complete semantics

Once an agent commits itself to a set of assumptions Δ, it is not unreasonable to expect that agent to accept any further assumption α which is "defended" by Δ, and then to accept any assumptions "defended" by $\Delta \cup \{\alpha\}$, etc. Repeatedly adding such assumptions to a set Δ eventually leads to a *complete* set of assumptions, which not only contains Δ, but also contains all the assumptions Δ "defends".

Definition 5.1. A set of assumptions Δ *defends* an assumption α if and only if for each closed set of assumptions Δ', if Δ' attacks α then Δ attacks $\Delta' - \Delta$.

Definition 5.2. Given an assumption-based framework $\langle T, Ab, \overline{\ } \rangle$ and a set of assumptions $\Delta \subseteq Ab$,

$$\mathit{Def}(\Delta) = \{\alpha \mid \Delta \text{ defends } \alpha\}.$$

The following theorem follows directly from the definitions:

Theorem 5.3. *A set of assumptions Δ is admissible if and only if*
- *Δ is closed, and*
- *$\Delta \subseteq \mathit{Def}(\Delta)$.*

Whereas a closed set of assumptions is *admissible* if and only if it is *contained* in the set of assumptions it defends, it is *complete* if and only if it is *identical* to the set of assumptions it defends:

Definition 5.4. A set of assumptions Δ is *complete* if and only if
- Δ is closed, and
- $\Delta = \mathit{Def}(\Delta)$.

It follows immediately from the definition that every complete set of assumptions is admissible. On the other hand, not every admissible set is complete. For example, in flat assumption-based frameworks $\emptyset$ is always admissible, but need not be complete. However

Theorem 5.5. *Every stable set of assumptions is complete.*

Proof. Assume Δ is stable. Since every stable set of assumptions is admissible, it suffices to show that Δ contains every assumption α it defends. If instead Δ defends $\alpha \notin \Delta$, then Δ also attacks α. So Δ attacks $\Delta - \Delta$, which is impossible. □

Although every stable set is complete, not every preferred set need be complete, as the following example shows.

Example 5.6. Consider the non-monotonic modal theory

$$\{\neg Lp \rightarrow q,\ \ \neg Lr \rightarrow \neg q\}.$$

The set of assumptions $\{\neg Lp\}$ is admissible, since it is closed, does not attack itself and attacks the only closed attack $\{\neg Lp, \neg Lr, \neg Lq, \ldots\} = Ab$ against it. Moreover, the assumption $\neg Lr$ is defended by $\{\neg Lp\}$. This can be seen by the fact that the only closed attack against $\neg Lr$ is again $\{\neg Lp, \neg Lr, \neg Lq, \ldots\}$ which is attacked by $\{\neg Lp\}$. However, $\{\neg Lp, \neg Lr\}$ is not admissible, since it attacks itself. Furthermore, from Theorem 4.6, we know that at least one preferred set of assumptions containing $\{\neg Lp\}$ exists. Call this set Δ. Then it is clear that $\neg Lr$ is also defended by Δ. It is also clear that $\neg Lr \notin \Delta$, because $\{\neg Lp, \neg Lr\}$ is not admissible. Hence Δ is not complete.

However, Corollary 5.8 of the following theorem states that in the case of flat assumption-based frameworks, every preferred set of assumptions is complete.

Theorem 5.7. *Let $\langle T, Ab, \overline{\ } \rangle$ be a flat assumption-based framework, $\Delta \subseteq Ab$ be admissible and $S \subseteq Ab$ be a set of assumptions defended by Δ (i.e. $S \subseteq Def(\Delta)$). Then $\Delta \cup S$ is also admissible.*

Proof. Let $\Delta' = \Delta \cup S$. Since $\langle T, Ab, \overline{\ } \rangle$ is flat, Δ' is closed. First we prove that Δ' attacks every attack against it. In fact, each attack against Δ' is either an attack against Δ, which is attacked by Δ (since Δ is admissible), or an attack against S, again attacked by Δ (since Δ defends S). Finally we prove that Δ' does not attack itself. In fact, if Δ' did attack itself, then Δ' would attack either Δ or S. In the first case, since Δ is admissible, Δ attacks Δ' and therefore S. Since Δ defends S, we have that Δ attacks the empty set of assumptions, which is impossible. In the second case, since Δ defends S, Δ attacks $\Delta' - \Delta = S - \Delta$. Again, since Δ defends S, we have that Δ attacks the empty set of assumptions, which is impossible. □

It follows immediately that

Corollary 5.8. *Every preferred set of assumptions of a flat assumption-based framework is complete.*

Complete sets of assumptions need not exist in general, as demonstrated by Example 5.6. However, it follows directly from the existence of at least a preferred set of assumptions for flat assumption-based frameworks and the above Corollary 5.8, that complete sets of assumptions always exist for flat assumption-based frameworks.

Complete sets generalise the notion of complete scenarios for logic programs as defined by Dung [10]:

Theorem 5.9. *For each logic program P and set of assumptions Δ in the assumption-based framework $\langle P, Ab, \overline{} \rangle$ corresponding to P, $P \cup \Delta$ is a complete scenario of P in the sense of* [10] *if and only if Δ is complete with respect to $\langle P, Ab, \overline{} \rangle$.*

From the equivalence (proved by [8]) between the stationary semantics [48] and complete scenarios semantics [29] of logic programs, it follows then that the notion of complete set of assumptions is equivalent to the stationary semantics.

6. Sceptical semantics

Until now we have focused our attention on various credulous semantics. We shall now investigate sceptical semantics. In general, we can define a sceptical semantics which accepts a conclusion if and only if the conclusion holds in every (credulously) "acceptable" extension, where "acceptability" is understood in terms of maximal conflict-free, stable, admissible, preferred or complete extensions. In this section we will investigate two sceptical semantics. The first is the sceptical version of the complete semantics, the second is the sceptical version of the naive semantics.

6.1. Well-founded semantics

The well-founded semantics of logic programming [60] is a sceptical semantics which accepts a conclusion if and only if it holds in all complete extensions. This leads to the following generalisation in our framework.

Definition 6.1. A set of assumptions Δ is *well-founded* if and only if Δ is the intersection of all complete sets of assumptions.

Note that, because *Def* is monotonic (see Definition 5.2), it possesses a unique least fixed point, which coincides with $\bigcup\{Def^i(\emptyset) \mid i \text{ is an ordinal number}\}$. If this fixed point is closed then it is (minimally) complete and therefore well-founded. This is guaranteed to be the case for flat assumption-based frameworks (see Definition 4.10):

Theorem 6.2. *For every flat assumption-based framework, the well-founded set of assumptions is minimal (with respect to set inclusion) complete and coincides with the least fixed point of the operator Def.*

Proof. Since the framework is flat, $\emptyset$ is admissible. From Theorem 5.7, it follows immediately that for each ordinal i, the set $\bigcup\{Def^i(\emptyset) \mid i \leqslant n$ and n is an ordinal number$\}$ is admissible. Therefore, the least fixed point of *Def*, $\bigcup\{Def^i(\emptyset) \mid i$ is an ordinal number$\}$, is admissible, and therefore does not attack itself (and is closed). Hence, it is complete and therefore well-founded. □

Therefore, for flat assumption-based frameworks, a well-founded, sceptical agent is willing to make default assumptions Δ but it is not willing to commit itself to Δ sufficiently to assume Δ in the course of defending Δ against attack. Rather, it restricts itself either to defending Δ without making any assumptions at all or to defending Δ with the aid of assumptions which can be justified without assuming Δ to start with.

From Theorem 6.2 it follows that the well-founded set of assumptions is complete for every logic program and default theory. Moreover, in the case of logic programming, this set corresponds to the well-founded semantics of Van Gelder, Ross and Schlipf [60]:

Theorem 6.3. *Let P be a normal logic program and $\langle P, Ab, \overline{}\rangle$ the corresponding assumption-based framework. Then $\Delta \subseteq Ab$ is well-founded with respect to $\langle T, Ab, \overline{}\rangle$ if and only if $\{p \mid P \cup \Delta \vdash p\} \cup \{\neg p \mid not\ p \in \Delta\}$ is the well-founded model of P.*

This theorem follows directly from the results shown in [10].

Note that Theorem 6.2 gives a bottom-up method for computing the well-founded semantics of a flat assumption-based framework by computing $\bigcup\{Def^i_{\mathcal{F}}(\emptyset) \mid i$ is an ordinal number$\}$.

The well-founded semantics is more sceptical than the semantics obtained by taking the intersection of all preferred or stable extensions, as implied by the following Theorems 6.4 and 6.5 and as illustrated by Example 6.6:

Theorem 6.4. *For every flat assumption-based framework, the well-founded set of assumptions is contained in every preferred set of assumptions.*

Proof. Note that the well-founded set of assumptions is complete for any flat assumption-based framework and is contained in every complete set by definition. Moreover, every preferred set of assumptions of a flat assumption-based framework is complete, by Theorem 5.8. □

It follows directly from this theorem and from Theorem 4.6 that

Theorem 6.5. *For every flat assumption-based framework, the well-founded set of assumptions is contained in every stable set of assumptions.*

Example 6.6. Let T be the logic program:

$$\{p \leftarrow not\ q,\ \ q \leftarrow not\ p,\ \ r \leftarrow p,\ \ r \leftarrow q\}$$

There are two stable sets of assumptions, $\{not\ p\}$ and $\{not\ q\}$, which coincide with the preferred sets of assumptions. The conclusion r is justified by both of them. The well-founded set of assumptions, however, is $\emptyset$, and does not justify r.

However, the well-founded set of assumption is not always contained in every admissible set of assumptions. In particular, the empty set of assumptions is always admissible for flat assumption-based frameworks, but need not to be well-founded. For this reason, the semantics obtained by taking the intersection of all admissible extensions is more sceptical than the well-founded semantics.

6.2. Circumscription

Whereas circumscription [38] is usually defined model-theoretically, we interpret circumscription syntactically, in terms of sets of assumptions, when every model is a Herbrand model. This is the case, for example, when the theory T contains no function symbols and satisfies unique names axioms and domain closure axioms.

When every model is a Herbrand model, circumscription is the sceptical version of Theorist. Whereas in Theorist the set of assumptions Ab can be any subset of $\mathcal{L}$, in our treatment of circumscription the set of assumptions Ab consists of ground literals (atoms and their negation) for predicates which are fixed and ground negative literals for predicates which are minimised.

More formally, let T be a theory in a first-order language $\mathcal{L}$. Let $\mathcal{P}$ be a set of predicate symbols of $\mathcal{L}$ whose interpretation is to be minimised, $\mathcal{Z}$ a set of predicate symbols of $\mathcal{L}$ whose interpretation is to be varied and $\mathcal{Q}$ the set of remaining predicate symbols of $\mathcal{L}$, whose interpretation is to be fixed. $Ab = \mathcal{HB}^{\mathcal{P}}_{\neg} \cup \mathcal{HB}^{\mathcal{Q}}_{\neg} \cup \mathcal{HB}^{\mathcal{Q}}$, where

- $\mathcal{HB}^{\mathcal{P}}_{\neg}$ is the set of all sentences of the form

  $$\neg p(t_1, \ldots, t_n)$$

  with $p \in \mathcal{P}$,
- $\mathcal{HB}^{\mathcal{Q}}_{\neg}$ is the set of all sentences of the form

  $$\neg q(t_1, \ldots, t_n)$$

  with $q \in \mathcal{Q}$,
- $\mathcal{HB}^{\mathcal{Q}}$ is the set of all sentences of the form

  $$q(t_1, \ldots, t_n)$$

  with $q \in \mathcal{Q}$,

and $t_1, \ldots, t_n$ are ground terms constructible from the vocabulary of $\mathcal{L}$.

We will show (whenever every model is a Herbrand model) that a sentence α follows from the circumscription of T if and only if α holds in all maximal conflict-free extensions of $\langle T, Ab, \overline{\ }\rangle$, where $\overline{\beta} = \neg\beta$ (so that conflict-freedom and consistency coincide).

In the standard formulation [32], a sentence follows from the *circumscription* of T, $CIRC[T; \mathcal{P}; \mathcal{Z}]$, minimising the interpretation of predicate symbols in $\mathcal{P}$ and allowing the interpretation of predicates symbols in $\mathcal{Z}$ to vary if and only if the sentence holds in all $(\mathcal{P}, \mathcal{Z})$-minimal models of T, defined as follows: Let M and N be models of T. Then, $N \leqslant_{\mathcal{P},\mathcal{Z}} M$ if and only if M and N differ only in the interpretation of $\mathcal{P}$ and $\mathcal{Z}$, and the interpretation of $\mathcal{P}$ in N is a subset of its interpretation in M. A model M of T is *$(\mathcal{P}, \mathcal{Z})$-minimal* if and only if, for every model N of T such that $N \leqslant_{\mathcal{P},\mathcal{Z}} M$, $M \leqslant_{\mathcal{P},\mathcal{Z}} N$.

Theorem 6.7. *If every model of T is a Herbrand model of T, then*

(1) *every $(\mathcal{P}, \mathcal{Z})$-minimal model M of T is a model of a maximal conflict-free extension of $\langle T, Ab, \overline{\ }\rangle$;*

(2) *every model of a maximal conflict-free extension of $\langle T, Ab, \overline{\ }\rangle$ is a $(\mathcal{P}, \mathcal{Z})$-minimal model of T.*

Proof. (1) Let M be a $(\mathcal{P}, \mathcal{Z})$-minimal Herbrand model of T. Let Δ be the set of assumptions $M^{\mathcal{P}}_{\neg} \cup M^{\mathcal{Q}}_{\neg} \cup M^{\mathcal{Q}}$, where, for $S = \mathcal{P}$ or $S = \mathcal{Q}$, $M^{S}_{\neg} = \{\neg\alpha \in \mathcal{HB}^{S}_{\neg} \mid \alpha \notin M\}$ and $M^{S} = \{\alpha \in \mathcal{HB}^{S} \mid \alpha \in M\}$. From the $(\mathcal{P}, \mathcal{Z})$-minimality of M, it is clear that $M^{\mathcal{P}}_{\neg}$ is maximal and therefore $T \cup M^{\mathcal{P}}_{\neg} \cup M^{\mathcal{Q}}_{\neg} \cup M^{\mathcal{Q}} \vdash M^{\mathcal{P}}$.

Therefore, $T \cup \Delta$ is maximal conflict-free. Moreover, by construction of Δ, M is obviously a model of $T \cup \Delta$.

(2) Let M be a model of a maximal conflict-free extension $T \cup \Delta$. From the maximality of $T \cup \Delta$, for every atom q in a predicate in $\mathcal{Q}$, either $q \in \Delta$ or $\neg q \in \Delta$. Further, again from the maximality of $T \cup \Delta$, for every atom p in a predicate in $\mathcal{P}$, if $\neg p \notin \Delta$ then $T \cup \Delta \vdash p$. Therefore, all models of $T \cup \Delta$ coincide on the extension of $\mathcal{P}$ and $\mathcal{Q}$. Assume now that M is not $(\mathcal{P}, \mathcal{Z})$-minimal. Then, there must be a model N such that $N \leqslant_{\mathcal{P},\mathcal{Z}} M$. Therefore, N and M coincide on the extension of $\mathcal{Q}$ and every $\mathcal{P}$-atom that is false in M is also false in N. Hence, N is also a model of $T \cup \Delta$ that does not coincide with M on the extension of $\mathcal{P}$. This is a contradiction. □

It follows directly from this theorem and from the definition of circumscription, that

Corollary 6.8. *If every model of T is a Herbrand model of T, then, for any sentence $\alpha \in \mathcal{L}$, α holds in $CIRC(T; \mathcal{P}; \mathcal{Z})$ if and only if α holds in all maximal conflict-free extensions of $\langle T, Ab, \overline{\ }\rangle$.*

If α in Corollary 6.8 is restricted to ground clauses, then the corollary still holds under more general conditions, for example when every model of T contains a submodel which is a Herbrand model. With this restriction on the sentences α, Corollary 6.8 also follows from Theorem 2.8 (or the equivalent Proposition 2.10) and Theorem 2.5 of [23], and Theorem 2.6 of [44]. A version of Corollary 6.8, where T satisfies uniqueness of names axioms (and equality axioms), domain closure axioms, $\mathcal{L}$ contains no function symbols and there are no fixed predicates has been proved by Ginsberg [22, corollary 2.2]. A more general version of Corollary 6.8 above has been proved by Poole [45, Theorem 4.5.1]. The if-half of Corollary 6.8 is related to Observation 3.4.11 in [35].

7. Existence, coincidence and uniqueness of semantics

In this section, we investigate two classes of flat assumption-based frameworks. We show that for the first class, stratified assumption-based frameworks, the well-founded semantics, which exists by Theorem 6.2, is also stable (and therefore preferred). Thus, for stratified frameworks, well-founded, preferred and stable semantics always exist, coincide, and are unique. We show that for the second class, order-consistent assumption-based frameworks, any preferred set of assumptions, which is guaranteed to exist by Corollary 4.11, is also stable. Thus, for order-consistent frameworks, preferred and stable semantics exist and coincide (but might not be unique).

Both classes of framework are characterised in terms of their attack relationship graphs.

Definition 7.1. The *attack relationship graph* of a flat assumption-based framework $\langle T, Ab, \overline{} \rangle$ is a directed graph whose nodes are the assumptions in Ab and such that there exists an edge from an assumption δ to an assumption α if and only if δ belongs to a minimal (with respect to set inclusion) attack Δ against α.

Definition 7.2. A flat assumption-based framework is *stratified* if and only if its attack relationship graph is well-founded, i.e. it contains no infinite path of the form $\alpha_1, \ldots, \alpha_n, \ldots$, where for every $n \geqslant 0$ there is an edge from α_{n+1} to α_n.

Example 7.3. The framework corresponding to the logic program

$$\{p \leftarrow not\ q,\ \ q \leftarrow not\ p\}$$

is not stratified, because its attack relationship graph has an infinite path:

$$not\ p, not\ q, \ldots, not\ p, not\ q, \ldots.$$

Example 7.4. The framework corresponding to the logic program

$$\{p(X) \leftarrow not\ p(s(X)),\ \ p(0)\}$$

is not stratified, because its attack relationship graph has an infinite path:

$$not\ p(0), not\ p(s(0)), not\ p(s(s((0)))), \ldots.$$

Theorem 7.5. *For any stratified assumption-based framework, there exists a unique stable set of assumptions, which coincides with the well-founded set of assumptions.*

Proof. We need to show only that the well-founded set of assumptions is stable. From this and from the fact that the well-founded set of assumptions is contained in every stable set, it follows that there exists a unique stable set, which coincides with the well-founded set of assumptions.

Let $\langle T, Ab, \overline{} \rangle$ be a stratified assumption-based framework and let $\Delta \subseteq Ab$ be the well-founded set of assumptions of $\langle T, Ab, \overline{} \rangle$. Trivially, Δ does not attack itself. Moreover, since $\langle T, Ab, \overline{} \rangle$ is flat, Δ is closed. It remains to show that Δ attacks every $\alpha \notin \Delta$.

Assume the contrary. We will construct an infinite sequence of assumptions $\alpha_0, \ldots, \alpha_n, \ldots$ such that

- for each $i \geqslant 0$, $\alpha_i \notin \Delta$ and α_i is not attacked by Δ,
- for each $i \geqslant 0$, there is an edge from α_{i+1} to α_i in the attack relationship graph,

contradicting the assumption that $\langle T, Ab, \overline{\ } \rangle$ is stratified.

First, from the assumption that Δ is not stable, it is clear that there exists an $\alpha_0 \notin \Delta$ such that α_0 is not attacked by Δ. Suppose we have already constructed a finite sequence $\alpha_0, \ldots, \alpha_n$ satisfying the above two properties. Then $\alpha_n \notin \Delta$. Therefore, Δ does not defend α_n, and there exists a minimal Δ' such that Δ' attacks α_n but Δ does not attack $\Delta' - \Delta$. So there exists $\alpha_{i+1} \in \Delta' - \Delta$ such that α_{i+1} is not attacked by Δ. It is clear that there exists an edge from α_{i+1} to α_i. □

There are meaningful frameworks which have a stable semantics but are not stratified, for example the framework corresponding to the logic program in Example 7.3. We will show that for the class of order-consistent frameworks, which contains this program, a stable semantics always exists.

We will call an assumption δ "hostile" to an assumption α if either it belongs to a minimal attack against α or it is hostile to an assumption which is friendly to α. An assumption β is "friendly" to α if β is α or β is hostile to an assumption δ which is hostile to α. An assumption δ is "two-sided" towards an assumption α if it is both hostile and friendly. Equivalently:

Definition 7.6. Given a flat assumption-based framework $\langle T, Ab, \overline{\ } \rangle$, let $\delta, \alpha \in Ab$.

- δ is *friendly to* α if and only if there exists in the attack relationship graph for $\langle T, Ab, \overline{\ } \rangle$ a path with an even number of edges from δ to α.
- δ is *hostile to* α if and only if in the attack relationship graph for $\langle T, Ab, \overline{\ } \rangle$ there exists a path with an odd number of edges from δ to α.
- δ is *two-sided to* α, written $\delta \prec \alpha$, if and only if in the attack relationship graph for $\langle T, Ab, \overline{\ } \rangle$ there exist both a path with an even number of edges and a path with an odd number of edges from δ to α.

Definition 7.7. A flat assumption-based framework $\langle T, Ab, \overline{\ } \rangle$ is *order-consistent* if the relation $\prec$ is well-founded, i.e. there exists no infinite sequence $\alpha_1, \ldots, \alpha_n, \ldots$ where for every $n \geqslant 0$, $\alpha_{n+1} \prec \alpha_n$.

Example 7.8. The framework corresponding to the logic program

$$\{p \leftarrow not\ p\}$$

is not order-consistent, because there exists an infinite sequence *not p*, ..., *not p*, ...

It is easy to see that

Theorem 7.9. *Every stratified assumption-based framework is order-consistent.*

Theorem 7.10. *For every order-consistent assumption-based framework stable sets of assumptions are preferred sets of assumptions and vice versa.*

Proof. Since every stable set of assumptions is preferred, we need to prove only that every preferred set of assumptions is stable. Let $\langle T, Ab, \overline{\ } \rangle$ be an order-consistent assumption-based framework and let $\Delta \subseteq Ab$ be a preferred set of assumptions which is not stable. We will construct an admissible set Δ_0 containing Δ as a proper subset, thereby contradicting the assumption that Δ is preferred.

Let $Ab' = Ab - (\Delta \cup \{\alpha \mid \Delta \text{ attacks } \alpha\})$. Since Δ is not stable, it is clear that Ab' is not empty.

Let $\alpha \in Ab'$ be such that there exists no $\beta \in Ab'$ such that $\alpha \succ \beta$. (The existence of such α is guaranteed by the order-consistency of the framework.)

Define S_0 (respectively S_1) to be the set consisting of all those $\beta \in Ab'$ such that there exists a path with an even (respectively odd) number of edges in the attack relationship graph from β to α. It follows from the definition of α that S_0 and S_1 are disjoint. Note that due to the definition of α, $\alpha \in S_0$. Hence $S_0 \neq \emptyset$.

Note that there exists at least one attack A against each $\beta \in S_1$ such that $\emptyset \neq A - \Delta \subseteq S_0$. (Otherwise either β is attacked by Δ or β is defended by Δ. Either way, $\beta \notin Ab'$.) This implies that $\Delta \cup S_0$ attacks each assumption in S_1.

Let $\Delta_0 = \Delta \cup S_0$. We want to show now that Δ_0 is admissible. Let Δ' be an attack against some assumption in S_0. If Δ attacks Δ' then there is nothing to prove. Assume now that Δ does not attack Δ'. Therefore $\emptyset \neq \Delta' - \Delta \subseteq Ab'$. Hence $\Delta' - \Delta \subseteq S_1$. Hence Δ_0 attacks $\Delta' - \Delta$. This means that Δ_0 attacks Δ'. Further, we show that Δ_0 does not attack itself. If Δ_0 did attack itself, then Δ_0 would attack Δ or S_0. If Δ_0 did attack S_0, then $Ab' \cap (\Delta_0 - \Delta) \subseteq S_1$. From $S_0 = \Delta_0 - \Delta$, it follows $S_0 \subseteq S_1$, which is impossible. If Δ_0 did attack Δ, then, since Δ is preferred, Δ would attack Δ, contradicting the assumption that Δ is preferred, or S_0, contradicting the assumption that $S_0 \subseteq Ab'$. Therefore Δ_0 is admissible and contains Δ as a proper subset. □

It follows directly from the definitions that the abstract notions of stratification and order-consistency generalise the notions of stratification and order-consistency for logic programming:

Theorem 7.11. *If P is a stratified logic program* [4], *then the corresponding assumption-based framework $\langle P, Ab, \overline{\ } \rangle$ is stratified. Similarly, if P is an order-consistent logic program* [51], *then the corresponding assumption-based framework $\langle P, Ab, \overline{\ } \rangle$ is order-consistent.*

8. Related work

The role of argumentation in human reasoning has been studied both inside and outside the field of artificial intelligence. Outside artificial intelligence, both Toulmin's [58] philosophical analysis of argumentation and Lorenz and Lorenzen's [34] logical analysis of classical logic as an argumentation game are particularly noteworthy.

Among the earliest investigations of argumentation in artificial intelligence, the work of Alvarado [2] and Birnbaum, Flowers and McGuire [3] focused on understanding the structure of arguments in editorials and political dialogues.

Pollock's work [41] bridges the fields of philosophy and artificial intelligence and, like this paper, addresses the use of argumentation for default reasoning. He constructs a theory of defeasible reasoning that takes into account the relations between arguments supporting contradictory conclusions. Dung [11] showed that Pollock's theory of defeasible reasoning corresponds to the computable part of the well-founded semantics of Section 6.1 in this paper. In his later work [42], Pollock develops an alternative, credulous semantics for defeasible reasoning. It is easy to see that this corresponds to the "stable theory" semantics for normal logic programs proposed by Kakas and Mancarella [28] and mentioned at the end of Section 4.

Simari and Loui [54] extend Pollock's sceptical semantics to incorporate Poole's formalisation of the principle that specific defaults have higher priority than more general defaults.

Vreeswijk [61] analyses different kinds of priorities that can arise between conflicting arguments and uses this analysis to decide how to resolve the conflict. But he does not develop this into a complete logic for default reasoning.

Touretzky, Horty and Thomason, [59] argue that Pollock's argumentation system cannot be used to formalise non-monotonic inheritance reasoning. Dung and Son [15] counterargue against [59] by showing that non-monotonic inheritance can be formalised using the argumentation-theoretic methods of this paper.

The approach to argumentation taken in this paper is most closely related to our earlier formalisations [6, 11], which were based upon the argumentation-theoretic interpretation of negation as failure in logic programming introduced by Kakas, Kowalski and Toni [25]. This was inspired, in part, by Dung's admissibility and preferred semantics [10] for logic programming, which was motivated, in turn, by Eshghi and Kowalski's abductive interpretation of stable model semantics [16, 17]. Dung subsequently formalised [11] argumentation in abstract terms, taking the notion of attack and argument as primitive.

In this paper, we revert to the approach taken in [25] and developed further in [6] in which assumptions are taken as primitive and both attacks and arguments are defined in terms of the monotonic derivability of conclusions based upon sets of assumptions.

Kakas [24] generalised the argumentation-theoretic interpretation of negation as failure and applied it to other logics for default reasoning. In particular, he proposed an argumentation-theoretic semantics for default logic different from the standard semantics and analogous to the acceptability semantics [27] for logic programming.

Toni and Kakas [55] develop abstract argumentation-theoretic proof procedures for computing admissibility, weak stability [28] and acceptability semantics [27] for default reasoning in general and normal logic programming in particular. In the companion paper [13] we show how an abstract proof procedure for the admissibility semantics can be derived systematically from its specification.

Recently, a number of authors have investigated other applications of argumentation to logic programming. Kakas, Mancarella and Dung [27] and Kakas and Dimopoulos [9] investigate argumentation-theoretic semantics and proof procedures for extended logic programs without negation as failure, but with priorities between clauses. Alferes and Pereira [1] use argumentation to expand the well-founded model of normal and extended logic programs. You and Cartwright [62] investigate the tractability of argumentation semantics for extended logic programming.

Independently of these developments, Geffner [18] shows that the well-founded semantics of logic programming can be understood in argumentation-theoretic terms. He also presents a bottom-up proof procedure for this semantics. Based upon Geffner's notion of argumentation, Torres [57] proposes an argumentation-theoretic semantics for negation as failure that is equivalent to Kakas and Mancarella's stable theory semantics [28].

Although our approach is based upon the abductive interpretation of negation as failure [16, 17] and Dung's admissibility and preferred semantics [10], it parallels many other approaches to argumentation developed independently in artificial intelligence. Among these, the work of Lin and Shoham's [33] is most clearly related to ours both in its aims and its methods.

Lin and Shoham [33] similarly develop an abstract argumentation-theoretic framework with the goal of capturing the semantics of many existing non-monotonic logics. They show that different variants of a single abstract notion of *complete* set of arguments corresponds to the standard semantics of default logic and autoepistemic logic. They also show a relationship to the semantics of stratified logic programs and the semantics of circumscription. Their notion of complete set of arguments is similar to our notion of stable set of assumptions.

Brewka and Konolige [7] also investigate default reasoning at a similar level of abstraction in abductive terms, but without employing an explicit notion of argument. They propose a new semantics, which they apply to a variety of non-monotonic logics, and which they argue improves upon the standard semantics of these logics.

Marek, Nerode and Remmel [36, 37] use their non-monotonic rule systems to provide an abstract framework to reconstruct the standard semantics of many non-monotonic logics. But they do not employ explicit notions of abduction or argumentation, and they do not consider the case of circumscription.

A number of other authors have employed argumentation for developing proof procedures rather than for semantics. Geffner and Pearl [19], for example, develop such a proof procedure for a conditional logic which has a sceptical model-theoretic semantics similar to circumscription. However, the proof procedure is incomplete for this semantics. we conjecture that the reason for this incompleteness may be that the proof procedure computes the well-founded semantics instead.

Ginsberg [22] and Baker and Ginsberg [5] develop an argumentation-theoretic proof procedure for circumscription. Like our argumentation-theoretic semantics of circumscription, their proof procedure is restricted to a case where arbitrary interpretations and Herbrand interpretations coincide.

Argumentation has become an important topic of research recently in the field of artificial intelligence and law. Prakken [46], for example, extends default logic using argumentation-theoretic notions to establish a preference between arguments based upon priorities between different default rules. Prakken and Sartor [47] formalise similar notions using the language of extended logic programs augmented with priorities. They extend Dung's [12] grounded semantics, which is a well-founded semantics for extended logic programs, to incorporate such priorities. Kowalski and Toni [31], on the other hand, argue that priorities can be dealt with by expressing the assumption that a rule is not defeated by a higher priority rule by means of an explicit condition of the rule rather than by dealing with priorities in the semantics.

9. Conclusions and future work

The abstract argumentation-theoretic semantics we developed in this paper shows that most formalisations of default reasoning can be viewed as extending a given theory by means of assumptions. In each case, these assumptions can be understood as expressing that their contraries cannot be shown. In most cases, the existing semantics sanction an extension if it is maximal conflict-free or if it is stable in the sense that it attacks all assumptions not in the extension. Many of these semantics are credulous, sanctioning a conclusion if it holds in some acceptable extension. Others are sceptical, sanctioning a conclusion if it holds in all acceptable extensions.

We have argued that stable semantics, which is the standard semantics of most formalisations of default reasoning, is too restrictive and have proposed admissibility semantics as an alternative. As we have remarked earlier, admissibility semantics can also be improved by generalising the stable theory semantics [28] and acceptability semantics [27] for logic programming of Kakas and Mancarella.

Admissibility semantics and its improvements have the further advantage over stable semantics that they can be implemented more easily by means of a natural refinement of the semantics. In a companion paper [13], we show how most proof procedure for such semantics can be derived from the semantics. For this purpose we formalise the proof procedure as a logic program and the semantics as a program specification. We use well established techniques for logic program synthesis and verification to derive the program from the specification.

We foresee three main research direction for the work presented in this paper:

(1) Other existing logics and other semantics for default reasoning can be investigated in argumentation-theoretic terms. In particular, it would be useful to determine whether any of the many existing proposals for improving the semantics of existing logics correspond to the admissibility semantics and its improvements.

(2) The abstract argumentation theoretic framework should be developed further with the aim of identifying other improvements. If possible, we should evaluate the different existing logics in argumentation terms with the aim of identifying the best features of the individual logics and incorporating them into a single formalism.

(3) The argumentation theory should be applied to other problems of practical reasoning in areas such as law. We are particularly interested in the possibility that argumentation can help to reconcile conflicts between different sets of hypotheses. Some preliminary thoughts of this kind have been presented in [30].

Acknowledgements

This research was supported by the Fujitsu Research Laboratories and by the EEC activity KIT011-LPKRR. The first author was partially supported by the grant 93-011-16016 of the Russian Foundation for Fundamental Research. The authors are grateful to Murray Shanahan for helpful discussions, and to Victor Marek and the referees for helpful comments.

References

[1] J.J. Alferes and L.M. Pereira, An argumentation-theoretic semantics based on non-refutable falsity, in: Dix, Pereira, Przymusinski, eds., *Proceedings ICLP'94 Workshop on Non-Monotonic Extensions of Logic Programming* (1994).

[2] S.J. Alvarado, Argument comprehension, in: S. Shapiro, ed., *Encyclopedia of Artificial Intelligence* 30-52.

[3] L. Birnbaum, M. Flowers and R. McGuire, Towards an artificial intelligence model of argumentation, in: *Proceedings AAAI-80*, Stanford, CA (1980) 313-315.

[4] K.R. Apt, H. Blair and A. Walker, Towards a theory of declarative knowledge, in: J. Minker, ed., *Foundations of Deductive Databases and Logic Programming* (Morgan Kaufmann, Los Altos, CA, 1988).

[5] A.B. Baker and M.L. Ginsberg, A theorem prover for prioritised circumscription, in: *Proceedings IJCAI-89*, Detroit, MI (1989) 463-467.

[6] A. Bondarenko, F. Toni and R.A. Kowalski, An assumption-based framework for non-monotonic reasoning, in: A. Nerode and L. Pereira, eds., *Proceedings 2nd International Workshop on Logic Programming and Non-Monotonic Reasoning* (MIT Press, Cambridge, MA, 1993) 171-189.

[7] G. Brewka and K. Konolige, An abductive framework for general logic programs and other non-monotonic systems, in: *Proceedings IJCAI-93*, Chambery, France (1993) 9-15.

[8] A. Brogi, E. Lamma, P. Mello and P. Mancarella, Normal logic programs as open positive programs, in: *Proceedings ICSLP-92* (1992).

[9] Y. Dimopoulos and A.C. Kakas, Logic programming without negation as failure. in: *Proceedings ILPS-95* (MIT Press, Cambridge, MA, 1995).

[10] P.M. Dung, An argumentation theoretic foundation of logic programming, *J. Logic Programming* **22** (1995) 151-177; a shortened version appeared as "Negation as hypothesis: an abductive foundation for logic programming", in: *Proceedings ICLP-91* (MIT Press, Cambridge, MA, 1991).

[11] P.M. Dung, The acceptability of arguments and its fundamental role in non-monotonic reasoning, logic programming and *n*-person games, *Artificial Intelligence* **77** (1995) 321-357.

[12] P.M. Dung, An argumentation semantics for logic programming with explicit negation. in: *Proceedings ICLP'93* (MIT Press, Cambridge, MA) 616-630.

[13] P.M. Dung, R.A. Kowalski and F. Toni, Argumentation-theoretic proof procedures for non-monotonic reasoning, Draft, 1996.

[14] P.M. Dung and P. Ruamviboonsuk, Well-founded reasoning with classical negation, in: A. Nerode, V.W. Marek and D. Subrahmanian, eds., *Proceedings 1st International Workshop on Logic Programming and Nonmonotonic Reasoning* (1991) 120-135.

[15] P.M. Dung and T.C. Son, Non-monotonic inheritance, argumentation and logic programming, in: V.W. Marek, A. Nerode and M. Truszczynski, eds., *Proceedings 3rd International Workshop on Logic Programming and Non-Monotonic Reasoning*, Lecture Notes in Artificial Intelligence **928** (Springer, Berlin, 1995) 316-329.

[16] K. Eshghi and R.A. Kowalski, Abduction through deduction, Technical Report, Imperial College, London (1988).

[17] K. Eshghi and R.A. Kowalski, Abduction compared with negation as failure, in: *Proceedings ICLP-89* (MIT Press, Cambridge, MA, 1989).

[18] H. Geffner, Beyond negation as failure, in: *Proceedings KR-91*, Cambridge, MA (1991) 218-229

[19] H. Geffner and J. Pearl, Conditional entailment: bridging to approaches to default reasoning, *Artificial Intelligence* **53** (1992) 209-244.

[20] M. Gelfond and V. Lifschitz, The stable model semantics for logic programming, in: *Proceedings ICSLP-88* (MIT Press, Cambridge, MA, 1988).

[21] M. Gelfond and V. Lifschitz, Logic programs with classical negation, in: D.H.D. Warren and P. Szeredi, eds., *Proceedings ICLP-90* (MIT Press, Cambridge, MA, 1990) 579-597.

[22] M.L. Ginsberg, A circumscriptive theorem prover, *Artificial Intelligence* **39** (1989) 209-230.

[23] K. Inoue and N. Helft, On theorem provers for circumscripton, in: *Proceedings LPC-90* (1990) 115-123.

[24] A.C. Kakas, Default reasoning via negation as failure, in: G. Lakemeyer and B. Nebel, eds., *Proceedings ECAI-92 Workshop on Foundations of Knowledge Representation and Reasoning*, Lecture Notes in Artificial Intelligence **810** (Springer, Berlin).

[25] A.C. Kakas, R.A. Kowalski and F. Toni, Abductive logic programming, *J. Logic and Computation* **2** (1993).

[26] A.C. Kakas, R.A. Kowalski and F. Toni, The role of abduction in logic programming, in: *Handbook of Logic in Artificial Intelligence and Logic Programming* **5** (Oxford University Press, Oxford, to appear).

[27] A.C. Kakas, P. Mancarella and P.M. Dung, The acceptability semantics for logic programs, in: P. Van Hentenryck, ed., *Proceedings ICLP-94* (MIT Press, Cambridge, MA, 1994) 504-519.

[28] A.C. Kakas and P. Mancarella, Stable theories for logic programs, in: *Proceedings ISLP-91* (MIT Press, Cambridge, MA, 1991).

[29] A.C. Kakas and P. Mancarella, Preferred extensions are partial stable models, *J. Logic Programming* **14** (1992).

[30] R.A. Kowalski and F. Toni, Argument and reconciliation, in: *Proceedings FGCS Workshop on Application of Logic Programming to Legal Reasoning*, Tokyo, Japan (1994).

[31] R.A. Kowalski and F. Toni, Abstract argumentation artificial intelligence and law, to appear.

[32] V. Lifschitz, Circumscription, in: D. Gabbay, C. Hogger and J.A. Robinson, eds., *Handbook of Logic in Artificial Intelligence and Logic Programming* **3** (Oxford University Press, Oxford, 1994) 297-352.

[33] F. Lin and Y. Shoham, Argument systems: A uniform basis for non-monotonic reasoning, in: *Proceedings KR-89*, Cambridge, MA (1989).

[34] P. Lorenzen and K. Lorenz, *Dialogische Logik* (Wissenschaftliche Buchgesellschaft, Darmstadt, 1977).

[35] D. Makinson, General patterns in non-monotonic reasoning, in: D. Gabbay, C. Hogger and J.A. Robinson, eds., *Handbook of Logic in Artificial Intelligence and Logic Programming*, Vol. 3 (Oxford University Press, Oxford) 1994) 35-110.

[36] W. Marek, A. Nerode and J. Remmel, A theory of non-monotonic rule systems I, *Ann. Math. Artif. Intell.* **1** (1990) 241-273.

[37] W. Marek, A. Nerode and J. Remmel, A theory of non-monotonic rule systems II, *Ann. Math. Artif. Intell.* **5** (1992) 229-263.

[38] J. McCarthy, Circumscription-A form of non-monotonic reasoning, *Artificial Intelligence* **13** (1980) 27-39.

[39] D. McDermott, Nonmonotonic logic II: non-monotonic modal theories, *J. ACM* **29** (1982).

[40] R. Moore, Semantical considerations on non-monotonic logic, *Artificial Intelligence* **25** (1985).

[41] J.L. Pollock, Defeasible reasoning, *Cognitive Sci.* **11** (1987) 481-518.

[42] J.L. Pollock, Justification and defeat, *Artificial Intelligence* **67** (1994) 377-407.

[43] D. Poole, A logical framework for default reasoning, *Artificial Intelligence* **36** (1988) 27-47.

[44] D. Poole, Explanation and prediction: an architecture for default and abductive reasoning, *Comput. Intell. J.* **5** (1989) 97-110.

[45] D. Poole, Default logic, in: D. Gabbay, C. Hogger and J.A. Robinson, eds., *Handbook of Logic in Artificial Intelligence and Logic Programming* **3** (Oxford University Press, Oxford, 1994) 189-215.

[46] H. Prakken, Logical tools for modelling legal argument, Ph.D. Thesis, Free University Amsterdam (1993).

[47] H. Prakken and G. Sartor, On the relation between legal language and legal argument: Assumptions, applicability and dynamic priorities, in: *Proceedings ICAIL-95* (1995) 1-10.

[48] T.C. Przymusinski, Semantics of disjunctive logic programs and deductive databases, in: *Proceedings DOOD-91* (1991).

[49] R. Reiter, A logic for default reasoning, *Artificial Intelligence* **13** (1980) 81–132.

[50] D. Saccà and C. Zaniolo, Stable models and non-determinism for logic programs with negation, in: *Proceedings ACM SIGMOD-SIGACT Symposium on Principles of Database Systems* (1990).

[51] T. Sato, Completed logic programs and their consistency, *J. Logic Programming* **9** (1990) 33–44.

[52] K. Satoh and N. Iwayama, A correct top-down proof procedure for general logic programs with integrity constraints, in: E. Lamma and P. Mello, eds., *Proceedings 3rd International Workshop on Extensions of Logic Programming*, Lecture Notes in Artificial Intelligence **660** (Springer, Berlin, 1992) 19–34.

[53] G. Shvarts, Autoepistemic modal logics, in: *Proceedings Third Conference on Theoretical Aspects of Reasoning about Knowledge*, Pacific Grove, CA (1990).

[54] G.R. Simari and R.P. Loui, A mathematical treatment of defeasible reasoning and its implementation, *Artificial Intelligence* **53** (1992) 125–157.

[55] F. Toni and A.C. Kakas, Computing the acceptability semantics, in: V.W. Marek, A. Nerode and M. Truszczynski, eds., *Proceedings 3rd International Workshop on Logic Programming and Non-monotonic Reasoning*, Lecture Notes in Artificial Intelligence **928** (Springer, Berlin, 1995) 401–415

[56] F. Toni and R.A. Kowalski, Reduction of abductive logic programs to normal logic programs, in: L. Sterling, ed., *Proceedings ICLP-95* (MIT Press, Cambridge, MA) 367–381.

[57] A. Torres, Negation as failure to support, in: A. Nerode and L. Pereira, eds., *Proceedings 2nd International Workshop on Logic Programming and Non-Monotonic Reasoning* (MIT Press, Cambridge, MA, 1993) 223–243.

[58] S. Toulmin, *The Uses of Arguments* (Cambridge University Press, Cambridge, MA, 1958).

[59] D.S. Touretzky, J.F. Horty and R.H. Thomason, A sceptic's managerie: conflictors, preemptors, reinstaters and zombies in non-monotonic inheritance, in: *Proceedings IJCAI-91*, Sydney (Morgan Kaufmann, Los Altos, CA, 1991) 478–483.

[60] A. Van Gelder, K.A. Ross and J.S. Schlipf, Unfounded sets and the well-founded semantics for general logic programs, in: *Proceedings ACM SIGMOD-SIGACT Symposium on Principles of Database Systems* (1988).

[61] G. Vreeswijk, The feasibility of defeat in defeasible reasoning, in: *Proceedings KR-91*, Cambridge, MA (Morgan Kaufmann, Los Altos, CA, 1991).

[62] J.H. You and R. Cartwright, Tractable argumentation semantics via iterative belief revision, in: *Proceedings ILPS-94* (MIT Press, Cambridge, MA, 1994) 239–253.