A Tutorial on Default Logics

GRIGORIS ANTONIOU

Griffith University

Default Logic is one of the most prominent approaches to nonmonotonic reasoning, and allows one to make plausible conjectures when faced with incomplete information about the problem at hand. Default rules prevail in many application domains such as medical diagnosis and legal reasoning.

Several variants have been developed over the past years, either to overcome some perceived deficiencies of the original presentation, or to realize somewhat different intuitions. This paper provides a tutorial-style introduction to some important approaches of Default Logic. The presentation is based on operational models for these approaches, thus making them more easily accessible to a broader audience, and more easily usable in practical applications.

Categories and Subject Descriptors: I.2.3 [**Artificial Intelligence**]: Deduction and Theorem Proving—*Nonmonotonic reasoning and belief revision*; I.2.4 [**Artificial Intelligence**]: Knowledge Representation Formalisms and Methods—*Representation languages*

General Terms: Languages, Theory

Additional Key Words and Phrases: Default Logic, nonmonotonic reasoning, operational models

1. INTRODUCTION: DEFAULT REASONING

When an intelligent system (either computer-based or human) tries to solve a problem, it may be able to rely on complete information about this problem, and its main task is to draw the correct conclusions using classical reasoning. In such cases, classical predicate logic may be sufficient.

However, in many situations the system has only *incomplete information* at hand, be it because some pieces of information are unavailable, or because it has to respond quickly and does not have the time to collect all relevant data. Classical logic indeed has the capacity to represent and reason with certain aspects of incomplete information. But there are occasions where additional information needs to be "filled in" to overcome the incompleteness, because certain decisions must be made. In such cases the system has to make some plausible conjectures, which in the case of default reasoning are based on rules of thumb, called *defaults*. For example, an emergency doctor has to make some conjectures about the most probable causes of the symptoms observed. Obviously, it would be inappropriate to await the results of possibly

Author's address: School of Computing & Information Technology, Griffith University, Brisbane, QLD 4111, Australia; email: ga@cit.gu.edu.au.

CONTENTS

extensive and time-consuming tests before beginning with treatment.

When decisions are based on assumptions, these may turn out to be wrong in the face of additional information that becomes available; i.e., medical tests may lead to a modified diagnosis. The phenomenon of having to take back some previous conclusions is called *nonmonotonicity*; it says that if a statement φ follows from a set of premises M and $M \subseteq M'$, φ does not necessarily follow from M'. Default Logic, originally presented in Reiter [1980], provides formal methods to support this kind of reasoning.

Default Logic is perhaps the most prominent method for nonmonotonic reasoning, basically because of the simplicity of the notion of a default, and because defaults prevail in many application areas. However, there exist several alternative design decisions which have led to variations of the initial idea; actually, we can talk about a *family of default reasoning methods* because they share the same foundations.

In this paper we present the motivations and basic ideas of some of the most important default logic variants, and compare them both with respect to interconnections and the fulfillment of some properties. When designing a tutorial, it is good to have some aims against which the final product can be tested. The particular aims of this tutorial paper are the following:

—To present the basic ideas of Default Logic to persons without prior knowledge in the area. Only basic understanding of classical logic is required.

—To equip the reader with skills and methods so that they can apply the concepts of Default Logic to concrete situations. This is achieved by the use of operational models which can be applied in a straightforward manner to examples without having to make any guesses, as is the case with the usual fixpoint or quasiinductive definitions.

—To give the reader a sense of the diversity of the topic.

The paper is organized as follows: Section 2 presents the basics of Reiter's Default Logic. Section 3 discusses properties and design decisions of this approach, and outlines some alternative intuitions on these issues. Sections 4–6 describe some basic default logic variants: Justified Default Logic [Lukaszewicz 1988], Constrained Default Logic [Schaub 1992], and approaches to prioritization [Brewka 1994]. In Section 7 we briefly discuss some further default logics, namely Cumulative Default Logic [Brewka 1991], Rational Default Logic [Mikitiuk and Truszczynski 1995], Disjunctive Default Logic [Gelfond et al. 1991], and weak extensions [Marek and Truszczynski 1993].

No prior knowledge of Default Logic is required since this is a tutorial paper. However, we assume that the reader is

familiar with notation and the basic concepts of classical logic.

2. DEFAULT LOGIC

2.1 The Notion of a Default

A rule used by football organizers in Germany might be: "A football game shall take place, unless there is snow in the stadium." This rule of thumb is represented by the default

$$\frac{football : \neg snow}{takesPlace}.$$

The interpretation of the default is as follows: If there is no information that there will be snow in the stadium, it is reasonable to assume $\neg snow$ and conclude that the game will take place (so preparations can proceed). But if there is a heavy snowfall during the night before the game is scheduled, then this assumption can no longer be made. Now we have definite information that there is snow, so we cannot assume $\neg snow$, therefore the default cannot be applied. In this case we need to refrain from the previous conclusion (the game will take place), so the reasoning is nonmonotonic.

Before proceeding with more examples, let us first explain why classical logic is not appropriate to model this situation. Of course, we could use the rule

$$football \wedge \neg snow \rightarrow takesPlace.$$

The problem with this rule is that we have to definitively establish that there will be no snow in the stadium before applying the rule. But that would mean that no game could be scheduled in the winter, which would create a revolution in Germany! It is important to understand the difference between having to know that it will not snow, and being able to *assume* that it will snow. Defaults support the drawing of conclusions based upon assumptions.

The same example could have been represented by the default

$$\frac{football : takesPlace}{takesPlace},$$

together with the classical rule $snow \rightarrow \neg takesPlace$. In case we know $snow$ then we can deduce $\neg takesPlace$ in classical logic, therefore we cannot assume $takesPlace$, as required by the default. In this representation, the default says "Football matches usually takes place," and *exceptions* to this rule are represented by classical rules such as the above one.

Defaults can be used to model *prototypical reasoning*, which means that most instances of a concept have some property. One example is the statement "Typically, children have (living) parents" which may be expressed by the default

$$\frac{child(X) : hasParents(X)}{hasParents(X)}.$$

A further form of default reasoning is *no-risk reasoning*. It concerns situations where we draw a conclusion even if it is not the most probable, because another decision could lead to a disaster. Perhaps the best example is the following main principle of justice in the Western cultures: "In the absence of evidence to the contrary assume that the accused is innocent." In default form:

$$\frac{accused(X) : innocent(X)}{innocent(X)}.$$

Defaults naturally occur in many application domains. Let us give an example from legal reasoning. According to German law, a foreigner is usually expelled if they have committed a crime. One of the exceptions to this rule concerns political refugees. This information is expressed by the default

$$\frac{criminal(X) \wedge foreigner(X) : expel(X)}{expel(X)}$$

in combination with the rule

$$politicalRefugee(X) \rightarrow \neg expel(X).$$

Hierarchies with exceptions are commonly used in biology. Here is a standard example:

Typically, molluscs are shell-bearers.
Cephalopods are molluscs.
Cephalopods are not shell-bearers.

It is represented by the default

$$\frac{mollusc(X) : shellBearer(X)}{shellBearer(X)}$$

together with the rule

$$cephalopod(X) \rightarrow mollusc(X)$$

$$\wedge \neg shellBearer(X).$$

Defaults can be used naturally to model the *Closed World Assumption* [Reiter 1978], which is used in database theory, algebraic specification, and logic programming. According to this assumption, an application domain is described by certain axioms (in form of relational facts, equations, rules, etc.) with the following understanding: a ground fact (that is, a nonparameterized statement about single objects) is taken to be false in the problem domain if it does not follow from the axioms. The closed world assumption has the simple default representation

$$\frac{true : \neg\varphi}{\neg\varphi}$$

for each ground atom φ. The explanation of the default is: if it is consistent to assume $\neg\varphi$ (which is equivalent to not having a proof for φ) then conclude $\neg\varphi$.

Further examples of defaults can be found in, say, Besnard [1989]; Etherington [1987b]; Lukaszewicz [1990]; and Poole [1994].

2.2 The Syntax of Default Logic

A *default theory* T is a pair (W, D) consisting of a set W of predicate logic formulae (called the *facts* or *axioms* of T) and a countable set D of defaults. A *default* δ has the form

$$\frac{\varphi : \psi_1, \ldots, \psi_n}{\chi}$$

where $\varphi, \psi_1, \ldots, \psi_n, \chi$ are closed predicate logic formulae, and $n > 0$. The formula φ is called the *prerequisite*, $\psi_1, \ldots, \psi_n$ the *justifications*, and χ the *consequent* of δ. Sometimes φ is denoted by $pre(\delta)$, $\{\psi_1, \ldots, \psi_n\}$ by $just(\delta)$, and χ by $cons(\delta)$. For a set D of defaults, $cons(D)$ denotes the set of consequents of the defaults in D. A default is called *normal* iff it has the form $\varphi : \psi/\psi$.

One point that needs some discussion is the requirement that the formulae in a default be ground. This implies that

$$\frac{bird(X) : flies(X)}{flies(X)}$$

is *not* a default according to the definition above. Let us call such rules of inference *open defaults*. An open default is interpreted as a default schema, meaning that it represents a *set of defaults* (this set may be infinite).

A *default schema* looks like a default, the only difference being that $\varphi, \psi_1, \ldots, \psi_n, \chi$ are arbitrary predicate logic formulae (i.e., they may contain free variables). A default schema defines a set of defaults, namely

$$\frac{\varphi\sigma : \psi_1\sigma, \ldots, \psi_n\sigma}{\chi\sigma}$$

for all ground substitutions σ that assign values to all free variables occur-

ring in the schema. That means free variables are interpreted as being universally quantified *over the whole default schema*. Given a default schema

$$\frac{bird(X) : flies(X)}{flies(X)}$$

and the facts $bird(tweety)$ and $bird(sam)$, the default theory represented is $(\{bird(tweety), bird(sam)\}, \{bird(tweety) : flies(tweety)/flies(tweety), bird(sam) : flies(sam)/flies(sam)\})$.

2.3 Informal Discussion of the Semantics

Given a default $\varphi : \psi_1, \ldots, \psi_n/\chi$, its informal meaning is the following:

> *If φ is known, and if it is consistent to assume $\psi_1, \ldots, \psi_n$, then conclude χ.*

In order to formalize this interpretation we must say in which context φ should be known, and with what $\psi_1, \ldots, \psi_n$ should be consistent. A first guess would be the set of facts, but this turns out to be inappropriate. Consider the default schema

$$\frac{friend(X, Y) \wedge friend(Y, Z) : friend(X, Z)}{friend(X, Z)}$$

which says "Usually my friends' friends are also my friends". Given the information $friend(tom, bob)$, $friend(bob, sally)$ and $friend(sally, tina)$, we would like to conclude $friend(tom, tina)$. But this is only possible if we apply the appropriate instance of the default schema to $friend(sally, tina)$ and $friend(tom,) sally\}$. The latter formula stems from a previous application of the default schema.[1] If we did not admit this intermediate step and used the original facts only, then we could not get the expected result.

Another example is the default theory $T = (W, D)$ with $W = \{green, aaaMember\}$ and $D = \{\delta_1, \delta_2\}$ with

$$\delta_1 = \frac{green : \neg likesCars}{\neg likesCars},$$

$$\delta_2 = \frac{aaaMember : likesCars}{likesCars}.$$

If consistency of the justifications was tested against the set of facts, then both defaults could be subsequently applied. But then we would conclude both $likesCars$ and $\neg likesCars$, which is a contradiction. It is unintuitive to let the application of default rules lead to an inconsistency, even if they contradict each other. Instead, if we applied the first default, and then checked application of the second with respect to the *current knowledge* collected so far, the second default would be blocked: from the application of the first default we know $\neg likesCars$, so it is not consistent to assume $likesCars$. After these examples, here is the formal definition:

> $\delta = \varphi : \psi_1, \ldots, \psi_n/\chi$ is *applicable to* a deductively closed set of formulae E iff $\varphi \in E$ and $\neg\psi_1 \notin E, \ldots, \neg\psi_n \notin E$.

The example of Greens and AAA members indicates that there can be several competing current knowledge bases which may be inconsistent with one another. The semantics of Default Logic will be given in terms of *extensions* that will be defined as the current knowledge bases satisfying some conditions. Intuitively, extensions represent possible world views which are based on the given default theories; they seek to extend the set of known facts with "reasonable" conjectures based on the available defaults. The formal definition will be given in the next subsection. Here we just collect some desirable properties of extensions.

[1] With other instantiations, of course.

—An extension E should include the set W of facts since W contains the certain information available: $W \subseteq E$.

—An extension E should be deductively closed because we do not want to prevent classical logical reasoning. Actually, we want to draw *more* conclusions, and that is why we apply default rules in addition. Formally: $E = Th(E)$, where Th denotes the deductive closure.

—An extension E should be *closed under the application of defaults in D* (formally: if $\varphi : \psi_1, \ldots, \psi_n / \chi \in D$, $\varphi \in E$ and $\neg\psi_1 \notin E, \ldots, \neg\psi_n \notin E$ then $\chi \in E$). That is, we do not stop applying defaults until we are forced to. The explanation is that there is no reason to stop at some particular stage if more defaults might be applied; extensions are *maximal* possible world views.

These properties are certainly insufficient because they do not include any "upper bound," that is, they don't provide any information about which formulae should be excluded from an extension. So we should require that an extension E be *minimal with respect to these properties*. Unfortunately, this requirement is still insufficient. To see this, consider the default theory $T = (W, D)$ with $W = \{aussie\}$ and $D = \{aussie : drinksBeer/drinksBeer\}$. Let $E = Th(\{aussie, \neg drinksBeer\})$. It is easily checked that E is minimal with the three properties mentioned above, but it would be highly unintuitive to accept it as an extension, since that would support the following argument: "If Aussies usually drink Beer and if somebody is an Aussie, then assume that she does not drink Beer."

2.4 An Operational Definition of Extensions

For a given default theory $T = (W, D)$ let $\Pi = (\delta_0, \delta_1, \ldots)$ be a finite or infinite sequence of defaults from D without multiple occurrences. Think of Π as a possible order in which we apply some defaults from D. Of course, we don't want to apply a default more than once within such a reasoning chain because no additional information would be gained by doing so. We denote the initial segment of Π of length k by $\Pi[k]$, provided the length of Π is at least k (from now on, this assumption is always made when referring to $\Pi[k]$). With each such sequence Π we associate two sets of first-order formulae, $In(\Pi)$ and $Out(\Pi)$:

—$In(\Pi)$ is $Th(W \cup \{cons(\delta) | \delta$ occurs in $\Pi\}$. So, $In(\Pi)$ collects the information gained by the application of the defaults in Π and represents the *current knowledge base* after the defaults in Π have been applied.

—$Out(\Pi) = \{\neg\psi | \psi \in just(\delta)$ for some δ occurring in $\Pi\}$. So, $Out(\Pi)$ collects formulae that should not turn out to be true, i.e., that should not become part of the current knowledge base even after subsequent application of other defaults.

Let us give a simple example. Consider the default theory $T = (W, D)$ with $W = \{a\}$ and D containing the following defaults:

$$\delta_1 = \frac{a : \neg b}{\neg b}, \qquad \delta_2 = \frac{b : c}{c}.$$

For $\Pi = (\delta_1)$ we have $In(\Pi) = Th(\{a, \neg b\})$ and $Out(\Pi) = \{b\}$. For $\Pi = (\delta_2, \delta_1)$ we have $In(\Pi) = Th(\{a, c, \neg b\})$ and $Out(\Pi) = \{\neg c, b\}$.

Up to now we have not assured that

the defaults in Π can be applied in the order given. In the example above, (δ_2, δ_1) cannot be applied in this order (applied according to the definition in the previous subsection). To be more specific, δ_2 cannot be applied, since $b \notin In(()) = Th(W) = Th(\{a\})$ which is the current knowledge before we attempt to apply δ_2. On the other hand, there is no problem with $\Pi = (\delta_1)$; in this case we say that Π is a *process of* T. Here is the formal definition:

—Π is called a *process of* T iff δ_k is applicable to $In(\Pi[k])$, for every k such that δ_k occurs in Π.

Given a process Π of T we define the following:

—Π is *successful* iff $In(\Pi) \cap Out(\Pi) = \emptyset$, otherwise it is *failed*.

—Π is *closed* iff every $\delta \in D$ that is applicable to $In(\Pi)$ already occurs in Π. Closed processes correspond to the desired property of an extension E being closed under application of defaults in D.

Consider the default theory $T = (W, D)$ with $W = \{a\}$ and D containing the following defaults

$$\delta_1 = \frac{a : \neg b}{d}, \delta_2 = \frac{true : c}{b}.$$

$\Pi_1 = (\delta_1)$ is successful but not closed since δ_2 may be applied to $In(\Pi_1) = Th(\{a, d\})$. $\Pi_2 = (\delta_1, \delta_2)$ is closed but not successful: both $In(\Pi_2) = Th(\{a, d, b\})$ and $Out(\Pi_2) = \{b, \neg c\}$ contain b. On the other hand, $\Pi_3 = (\delta_2)$ is a closed and successful process of T. According to the following definition, which was first introduced in Antoniou and Sperschneider [1994], $In(\Pi_3) = Th(\{a, b\})$ is an extension of T, in fact its single extension.

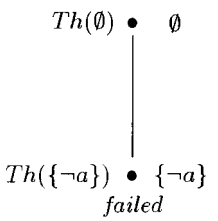

Figure 1. A process tree.

Definition 1. A set of formulae E is an *extension of* the default theory T iff there is some closed and successful process Π of T such that $E = In(\Pi)$.

In examples, it is often useful to arrange all possible processes in a canonical manner within a tree, called the *process tree* of the given default theory T. The nodes of the tree are labeled with two sets of formulae, an In-set (to the left of the node) and an Out-set (to the right of the node). The edges correspond to default applications, and are labeled with the default that is being applied. The paths of the process tree starting at the root correspond to processes of T.

2.5 Some Examples

Let $T = (W, D)$ with $W = \emptyset$ and $D = \{true : a/\neg a\}$. The process tree in Figure 1 shows that T has no extensions. Indeed, the default may be applied because there is nothing preventing us from assuming a. But when the default is applied, the negation of a is added to the current knowledge base, so the default invalidates its own application because both the In and the Out-set contain $\neg a$. This example demonstrates that there need not always be an extension of a default theory.

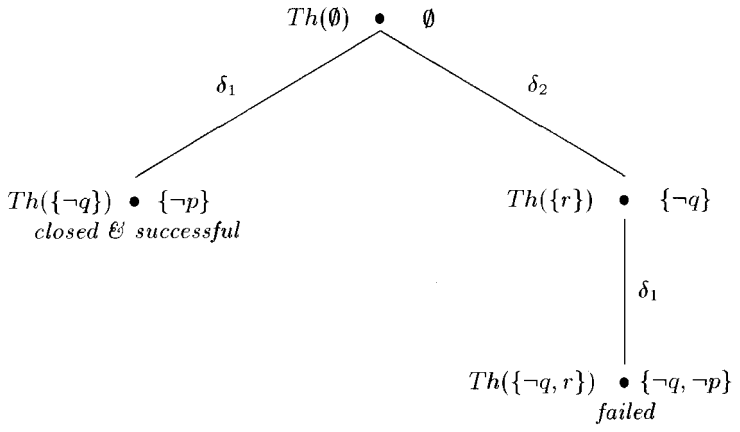

Figure 2. A process tree.

Let $T = (W, D)$ be the default theory with $W = \emptyset$ and $D = \{\delta_1, \delta_2\}$ with

$$\delta_1 = \frac{true : p}{\neg q}, \delta_2 = \frac{true : q}{r}.$$

The process tree of T is found in Figure 2 and shows that T has exactly one extension, namely $Th(\{\neg q\})$. The right path of the tree shows an example where application of a default destroys the applicability of a previous default: δ_1 can be applied after δ_2, but then $\neg q$ becomes part of the In-set, whilst it is also included in the Out-set (as the negation of the justification of δ_2).

Let $T = (W, D)$ with $W = \{green, aaaMember\}$ and $D = \{\delta_1, \delta_2\}$ with

$$\delta_1 = \frac{green : \neg likesCars}{\neg likesCars},$$

$$\delta_2 = \frac{aaaMember : likesCars}{likesCars}.$$

The process tree in Figure 3 shows that T has exactly two extensions (where g stands for *green*, a for *aaaMember*, and l for *likesCars*).

2.6 Reiter's Original Definition of Extensions

In this subsection we present Reiter's original definition of extensions [Reiter 1980]. In subsection 2.3 we briefly explained that the most difficult problem in describing the meaning of a default is to determine the appropriate set with which the justifications of the defaults must be consistent. The approach adopted by Reiter is to use some theory *beforehand*. That is, choose a theory which plays the role of a *context* or *belief set* and always check consistency against this context. Let us formalize this notion:

—A default $\delta = \varphi : \psi_1, \ldots, \psi_n / \chi$ *is applicable to* a deductively closed set of formulae F *with respect to belief set* E (the aforementioned context) iff $\varphi \in F$, and $\neg\psi_1 \notin E, \ldots, \neg\psi_n \notin E$ (that is, each ψ_i is consistent with E).

Note that the concept " δ is applicable to E" used so far is a special case where $E = F$. The next question that arises is which context to use. First, note that when a belief set E has been established, some formulae will become part of the knowledge base by applying de-

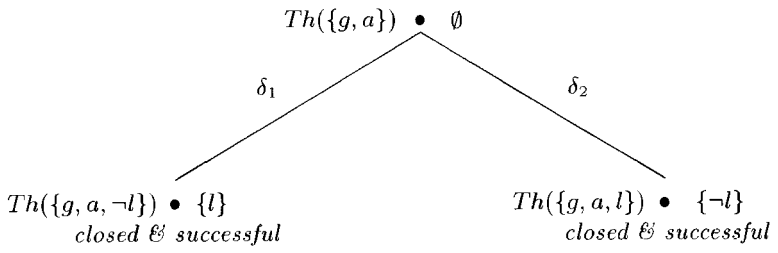

Figure 3. A process tree.

faults with respect to E. Therefore, they should be believed, i.e., be members of E. On the other hand, what would be a justification for a belief if it were not obtained from default application w.r.t. E? We require that E contain only formulae that can be derived from the axioms by default application w.r.t. E.

Let us now give a formal presentation of these ideas. For a set D of defaults, we say that F *is closed under* D *with respect to belief set* E iff, for every default δ in D that is applicable to F with respect to belief set E, its consequent χ is also contained in F.

Given a default theory $T = (W, D)$ and a set of formulae E, let $\Lambda_T(E)$ be the least set of formulae that contains W, is closed under logical conclusion (i.e., first-order deduction), and closed under D with respect to E. Informally speaking, $\Lambda_T(E)$ is the set of formulae that are sanctioned by the default theory T with respect to the belief set E.

Now, according to Reiter's definition, E is an extension of T iff $E = \Lambda_T(E)$. This fixpoint definition says that E is an extension iff by deciding to use E as a belief set, exactly the formulae in E will be obtained from default application. But please note the difficulty in applying this definition: we have to *guess* E and subsequently check for the fulfillment of the fixpoint equation. Having to guess is one of the most serious obstacles in understanding the concepts of Default Logic and in being able to apply them to concrete cases.

The following theorem shows that Reiter's extension concept is equivalent to the definition in subsection 2.4.

THEOREM 1. *Let* $T = (W, D)$ *be a default theory.* E *is an extension of* T *(in the sense of definition 2.1) iff* $E = \Lambda_T(E)$.

We conclude by giving a quasiinductive characterization of extensions, also due to Reiter: Given a default theory $T = (W, D)$, we say that E *has a quasiinductive definition in* T iff $E = \cup_i E_i$, where $E_0 = Th(W)$ and $E_{i+1} = Th(E_i \cup \{cons(\delta)\} \delta \in D)$ is applicable to E_i w.r.t. belief set $E\}$.

THEOREM 2. E *is an extension of* T *iff* E *has a quasiinductive definition in* T.

This characterization replaces the Λ_T-operator by a construction, both of them using the set E as context or belief set. Given a set of formulae E, this characterization is intuitively appealing. But notice that still it is necessary to first *guess* E before checking whether it is an extension. In this sense, the characterization is not as easy to apply as the process model from subsection 2.4.

The relationship of processes to the quasiinductive definition is that the tra-

versal of the process tree operationalizes the idea of guessing. More formally: if a branch of the process tree leads to a closed and successful process Π, then the quasiinductive construction using $In(\Pi)$ as a belief set yields the same result. But some branches of the process tree can lead to failed processes; this is the price we have to pay if we wish to avoid guessing.

3. PROPERTIES OF DEFAULT LOGIC

In this section we discuss some properties of Default Logic. Some of these properties can be interpreted as deficiencies, or they highlight some of Reiter's original "design decisions" and show alternative ideas that could be followed instead. In this sense, the discussion in this section motivates alternative approaches that will be presented in subsequent sections. One point that should be stressed is that there is not a "correct" default logic approach, but rather the most appropriate for the concrete problem at hand. Different intuitions lead to different approaches that may work better for some applications and worse for others.

3.1 Existence of Extensions

We saw that a default theory may not have any extensions. Is this a shortcoming of Default Logic? One might hold the view that if the default theory includes "nonsense" (for example $true : p/\neg p$), then the logic should indeed be allowed to provide no answer. According to this view, it is up to the user to provide meaningful information in the form of meaningful facts and defaults; after all, if a program contains an error, we don't blame the programming language.

The opposite view regards nonexistence of extensions as a drawback, and would prefer a more "fault-tolerant" logic; one which works even if some pieces of information are deficient. This viewpoint is supported by the trend towards heterogeneous information sources, where it is not easy to identify which source is responsible for the deficiency, or where the single pieces of information are meaningful, but lead to problems when put together.

A more technical argument in favor of the second view is the concept of *semi-monotonicity*. Default Logic is a method for performing nonmonotonic reasoning, so we cannot expect it to be monotonic when new knowledge is added to the set of facts. However, we might expect that the addition of new defaults would yield more, and not less information.[2] Formally, semi-monotonicity means the following:

> Let $T = (W, D)$ and $T' = (W, D')$ be default theories such that $D \subseteq D'$. Then for every extension E of T there is an extension E' of T' such that $E \subseteq E'$.

Default Logic violates this property. For example, $T = (\emptyset, \{true : p/p\})$ has the single extension $E = Th(\{p\})$, but $T' = (\emptyset, \{true : p/p, true : q/\neg q\})$ has no extension. So nonexistence of extensions leads to the violation of semi-monotonicity. Even though the concept of semi-monotonicity is not equivalent to the existence of extensions, these two properties usually come together (for a more formal support of this claim see Antoniou et al. [1996]).

If we adopt the view that the possible nonexistence of extensions is a problem, then there are two alternative solutions. The first one consists in restricting attention to those classes of default theories for which the existence of extensions is guaranteed. In his classical paper [Reiter 1980], Reiter already showed that if all defaults in a theory T are normal (in which case T is called a *normal default theory*), then T has at

[2]Some researchers would disagree with this view and regard semi-monotonicity as not desirable; see, for example, Brewka [1991].

least one extension. Essentially this is because all processes are successful, as can be easily seen.

THEOREM 3. *Normal default theories always have extensions. Furthermore, they satisfy semi-monotonicity.*

One problem with the restriction to normal default theories is that their expressiveness is limited. In general, it can be shown that normal default theories are strictly less expressive than general default theories. Normal defaults have limitations, particularly regarding the interaction among defaults. Consider the example

> Bill is a high school dropout.
> Typically, high school dropouts are adults.
> Typically, adults are employed.

These facts are naturally represented by the normal default theory $T = (\{dropout(bill)\}, \{dropout(X) : adult(X)/adult(X), adult(X) : employed(X)/employed(X)\})$. T has the single extension $Th(\{dropout(bill), adult(bill), employed(bill)\})$. It is acceptable to assume that Bill is adult, but it is counterintuitive to assume that Bill is employed! That is, whereas the second default *on its own* is accurate, we want to prevent its application in case the adult X is a high school dropout. This can be achieved if we change the second default to

$$\frac{adult(X) : employed(X) \wedge \neg dropout(X)}{employed(X)}.$$

But this default is not normal.[3] Defaults of this form are called *semi-normal*; Etherington [1987a] studied this class of default theories, and gave a sufficient condition for the existence of extensions. Another way of expressing interactions among defaults is the use of explicit priorities; this approach will be further discussed in Section 6.

Instead of imposing restrictions on the form of defaults in order to guarantee the existence of extensions, the other principal way is to modify the concept of an extension in such a way that all default theories have at least one extension, and that semi-monotonicity is guaranteed. In Sections 4 and 5 we will discuss two important variants with these properties, Lukaszewicz' Justified Default Logic and Schaub's Constrained Default Logic.

3.2 Joint Consistency of Justifications

It is easy to see that the default theory consisting of the defaults $true : p/q$ and $true : \neg p/r$ has the single extension $Th(\{q, r\})$. This shows that the *joint consistency* of justifications is not required. Justifications are *not* supposed to form a consistent set of beliefs, rather they are used to sanction "jumping" to some conclusions.

This design decision is natural and makes sense for many cases, but can also lead to unintuitive results. As an example consider the default theory, due to Poole, which says that, by default, a robot's arm (say a or b) is usable unless it is broken; further, we know that either a or b is broken. Given this information, we would not expect both a and b to be usable.

Let us see how Default Logic treats this example. Consider the default theory $T = (W, D)$ with $W = \{broken(a) \vee broken(b)\}$ and D consisting of the defaults

$$\frac{true : usable(a) \wedge \neg broken(a)}{usable(a)},$$

$$\frac{true : usable(b) \wedge \neg broken(b)}{usable(b)}.$$

[3]Note that it is unreasonable to add $\neg dropout(X)$ to the prerequisite of the default to keep it normal, because then we would have to definitely know that an adult is not a high school dropout before concluding that the person is employed.

Since we do not have definite information that a is broken we may apply the first default and obtain $E' = Th(W \cup \{usable(a)\})$. Since E' does not include $broken(b)$ we may apply the second default and get $Th(W \cup \{usable(a), usable(b)\})$ as an extension of T. This result is undesirable, as we know that either a or b is broken.

In Section 5 we shall discuss Constrained Default Logic as a prototypical Default Logic approach that enforces joint consistency of justifications of defaults involved in an extension.

The joint consistency property gives up part of the expressive power of default theories: under this property any default with several justifications $\varphi : \psi_1, \ldots, \psi_n / \chi$ is equivalent to the modified default $\varphi : \psi_1 \wedge \ldots \wedge \psi_n / \chi$ which has one justification. This is in contrast to a result in Besnard [1989], which shows that in Default Logic, defaults with several justifications are strictly more expressive than defaults with just one justification. Essentially, in default logics adopting joint consistency it is impossible to express default rules of the form "In case I am ignorant about p (meaning that I know neither p nor $\neg p$) I conclude q". The natural representation in default form would be $true : p, \neg p / q$, but this default can never be applied if joint consistency is required, because its justifications contradict one another; on the other hand it can be applicable in the sense of Default Logic.

Another example for which joint consistency of justifications is undesirable is the following.[4] When I prepare for a trip then I use the following default rules:

> If I may assume that the weather will be bad I'll take my sweater.
> If I may assume that the weather will be good then I'll take my swimsuit.

In the absence of any reliable information about the weather I am cautious enough to take both with me. But note that I am not building a consistent belief set upon which I make these decisions; obviously the assumptions of the default rules contradict each other. So Default Logic will treat this example in the intended way, whereas joint consistency of justifications will prevent me from taking both my sweater and my swimsuit with me.

3.3 Cumulativity and Lemmas

Cumulativity is, informally speaking, the property that allows for the safe use of lemmas. Formally: Let D be a fixed, countable set of defaults. For a formula φ and a set of formulae W we define $W \vdash_D \varphi$ iff φ is included in all extensions of the default theory (W, D). Now, cumulativity is the following property:

$$\textit{If } W \vdash_D \varphi, \textit{ then for all } \psi:$$

$$W \vdash_D \psi \Leftrightarrow W \cup \{\varphi\} \vdash_D \psi.$$

If we interpret φ as a lemma, cumulativity says that the same formulae can be obtained from W as from $W \cup \{\varphi\}$. This is the standard basis of using lemmas in, say, mathematics. Default Logic does not respect cumulativity: consider $T = (W, D)$ with $W = \emptyset$ and D consisting of the defaults

$$\frac{true : a}{a}, \frac{a \vee b : \neg a}{\neg a}$$

(this example is due to Makinson). The only extension of T is $Th(\{a\})$. Obviously, $W \vdash_D a$. From $a \vee b \in Th(\{a\})$ we get $W \vdash_D a \vee b$. If we take $W' = \{a \vee b\}$, then the default theory (W', D) has two extensions, $Th(\{a\})$ and $Th(\{\neg a, b\})$; therefore $W \cup \{a \vee b\} \not\vdash_D a$.

An analysis of cumulativity and other abstract properties of nonmonotonic inference is found in Makinson [1994]. A

[4] My thanks go to an anonymous referee.

lot of work has been invested in developing default logics that possess the cumulativity property, one notable approach being Brewka's Cumulative Default Logic [Brewka 1991]. But it is doubtful whether this is the right way to go, since it has additional conceptual and computational load, due to the use of assertions rather than plain formulae (see Section 7.2 for more information).

One might argue that semimonotonicity is rather unintuitive because it requires a *defeasible* conclusion which was based on some assumptions to be represented by a *certain* piece of information, that means a fact, and yet exhibit the same behaviour.

From the practical point of view the really important issue is whether we are able to represent and use lemmas in a safe way. How can we do this in Default Logic? Schaub [1992] proposed the representation of a lemma by a corresponding *lemma default* which records in its justifications the assumptions on which a conclusion was based. The formal definition of a lemma default is as follows.

Let Π_χ be a nonempty, successful process of T, minimal with the property $\chi \in In(\Pi_\chi)$. A *lemma default* δ_χ *corresponding to* χ is the default

$$\frac{true : \psi_1, \ldots, \psi_n}{\chi}$$

where $\{\psi_1, \ldots, \psi_n\} = \{\psi | \psi \in just(\delta)\}$ for a δ occurring in Π_χ. This default collects all assumptions that were used in order to derive χ.

THEOREM 4. *Let χ be included in an extension of T and δ_χ a corresponding lemma default. Then every extension of T is an extension of $T' = (W, D \cup \{\delta_\chi\})$, and conversely.*

So it is indeed possible to represent lemmas in Default Logic, not as facts (as required by cumulativity) but rather as defaults, which appears more natural anyway, since it highlights the nature of a lemma as having been proven defeasibly and thus as being open to disputation.

4. JUSTIFIED DEFAULT LOGIC

4.1 Motivation and Formal Presentation

Lukaszewicz considered the possible nonexistence of extensions as a representational shortcoming of the original Default Logic, and presented a variant, Justified Default Logic [Lukaszewicz 1988] which avoids this problem. The essence of his approach is the following: If we have a successful but not yet closed process, and all ways of expanding it by applying a new default lead to a failed process, then we stop and accept the current In-set as an extension. In other words, we take back the final, "fatal" step that causes failure.

Consider the default theory $T = (W, D)$ with $W = \{holidays, sunday\}$ and D consisting of the defaults

$$\delta_1 = \frac{sunday : goFishing \wedge \neg\, wakeUpLate}{goFishing},$$

$$\delta_2 = \frac{holidays : wakeUpLate}{wakeUpLate}.$$

It is easily seen that T has only one extension (in the sense of Section 2), namely $Th(\{holidays, sunday, wakeUpLate\})$. But if we apply δ_1 first, then δ_2 can be applied and leads to a failed process. In this sense we lose the intermediate information $Th(\{holidays, sunday, goFishing\})$. On the other hand, in Justified Default Logic we would stop after the application of δ_1 instead of applying δ_2 and running into failure; therefore we accept $Th(\{holidays, sunday, goFishing\})$ as an additional (modified) extension.

Technically, this is achieved by paying attention to maximally successful processes. Let T be a default theory,

and let Π and Γ be processes of T. We define $\Pi < \Gamma$ iff the set of defaults occurring in Π is a proper subset of the defaults occurring in Γ. Π is called a *maximal process of* T, iff Π is successful and there is no successful process Γ such that $\Pi < \Gamma$. A set of formulae E is called a *modified extension* of T iff there is a maximal process Π of T such that $E = In(\Pi)$.

In the example above, $\Pi = (\delta_1)$ is a maximal process: the only process that strictly includes Π is $\Gamma = (\delta_1, \delta_2)$ which is not successful. Therefore $Th(\{holidays, sunday, goFishing\})$ is a modified extension of T. T has another modified extension, which is the single extension $Th(\{holidays, sunday, wakeUpLate\})$ of T. Obviously every closed and successful process is a maximal process (since no new default can be applied). Therefore we have the following result:

THEOREM 5. *Every extension of a default theory T is a modified extension of T.*

In the process of a default theory T maximal processes correspond either to closed and successful nodes, or to nodes n such that all immediate children of n are failed.

It is instructive to look at a default theory without an extension, for example $T = (W, D)$ with $W = \emptyset$ and $D = \{true : p/\neg p\}$. The empty process is maximal (though not closed), because the application of the "strange default" would lead to a failed process, therefore $Th(\emptyset)$ is a modified extension of T. Since any branch of the process tree can be extended successfully to a modified extension, the following result can be shown.

THEOREM 6. *Every default theory has at least one modified extension. Furthermore, Justified Default Logic satisfies semi-monotonicity.*

4.2 Lukaszewicz' Original Definition

The original definition given in Lukaszewicz [1988] was based on fixpoint equations. Let $T = (W, D)$ be a default theory, and E, F, E' and F' sets of formulae. We say that a default $\delta = \varphi : \psi_1, \ldots, \psi_n/\chi$ is *applicable to* E' *and* F' *with respect to* E *and* F iff $\varphi \in E'$ and $E \cup \{\chi\} \models \neg\psi$ for all $\psi \in F \cup \{\psi_1, \ldots, \psi_n\}$.

E' and F' are *closed under the application of defaults in D with respect to E and F* iff, whenever a default $\delta = \varphi : \psi_1, \ldots, \psi_n/\chi$ in D is applicable to E' and F' with respect to E and F, $\chi \in E'$ and $\{\psi_1, \ldots, \psi_n\} \subseteq F'$.

Define $\Lambda_T^1(E, F)$ and $\Lambda_T^2(E, F)$ to be the smallest sets of formulae such that $\Lambda_T^1(E, F)$ is deductively closed, $W \subseteq \Lambda_T^1(E, F)$, and $\Lambda_T^1(E, F)$ and $\Lambda_T^2(E, F)$ are closed under D with respect to E and F.

The following theorem shows that modified extensions correspond exactly to sets E and F satisfying the fixedpoint equations $E = \Lambda_T^1(E, F)$ and $F = \Lambda_T^2(E, F)$. This is not surprising: intuitively, the idea behind the complicated definition of the Λ-operators is to maintain the set of justifications of defaults that have been applied (i.e., the sets F and F' which, in fact, correspond to $\neg Out(\Pi)$), and to avoid applications of defaults if they lead to an inconsistency with one of these justifications.

THEOREM 7. *Let T be a default theory. For every modified extension E of T there is a set of formulae F such that* $E = \Lambda_T^1(E, F)$ *and* $F = \Lambda_T^2(E, F)$.

Conversely, let E and F be sets of formulae such that $E = \Lambda_T^1(E, F)$ *and* $F = \Lambda_T^2(E, F)$. *Then E is a modified extension of T.*

5. CONSTRAINED DEFAULT LOGIC

5.1 Motivation and Definition

Justified Default Logic avoids running into inconsistencies and can therefore guarantee the existence of modified extensions. On the other hand, it does not require joint consistency of default justifications; for example, the default theory $T = (W, D)$ with $W = \emptyset$ and $D = \{true : p/q, true : \neg p/r\}$ has the single modified extension $Th(\{q, r\})$. Constrained Default Logic [Schaub 1992; Delgrande et al. 1994] is a Default Logic approach which enforces joint consistency. In the example above, after the application of the first default the second default may not be applied because p contradicts $\neg p$.

Furthermore, since the justifications are consistent with each other, we test the consistency of their conjunction with the current knowledge base. In the terminology of processes, we require the consistency of $In(\Pi) \cup \neg Out(\Pi)$.

Finally, we adopt the idea from the previous section that namely a default may only be applied if it does not lead to a contradiction (failure) a posteriori. That means, if $\varphi : \psi_1, \ldots, \psi_n/\chi$ is tested for application to a process Π, then $In(\Pi) \cup \neg Out(\Pi) \cup \{\psi_1, \ldots, \psi_n, \chi\}$ must be consistent. We note that the set Out no longer makes sense since we require joint consistency. Instead, we have to maintain the set of formulae which consists of W, all consequents *and all justifications* of the defaults that have been applied.

—Given a default theory $T = (W, D)$ and a sequence Π of defaults in D without multiple occurrences, we define $Con(\Pi) = Th(W \cup \{\varphi\} | \varphi$ is the consequent or a justification of a default occurring in $\Pi\}$. Sometimes we refer to $Con(\Pi)$ as the *set of constraints* or the *set of supporting beliefs*.

$Con(\Pi)$ represents the set of beliefs supporting Π. For the default theory $T = (W, D)$ with $W = \emptyset$ and $D = \{\delta_1 = true : p/q, \delta_2 = true : \neg p/r\}$ let $\Pi_1 = (\delta_1)$. Then $Con(\Pi_1) = Th(\{p, q\})$.

We say that a default $\delta = \varphi : \psi_1, \ldots, \psi_n/\chi$ *is applicable to a pair of deductively closed sets of formulae* (E, C) iff $\varphi \in E$ and $\psi_1 \wedge \ldots \wedge \psi_n \wedge \chi$ is consistent with C. A pair (E, C) of deductively closed sets of formulae is called *closed under* D if, for every default $\varphi : \psi_1, \ldots, \psi_n/\chi \in D$ that is applicable to (E, C), $\chi \in E$ and $\{\psi_1, \ldots, \psi_n, \chi\} \subseteq C$.

In the example above, δ_2 is not applicable to $(In(\Pi_1), Con(\Pi_1)) = (Th(\{q\}),)$ $Th(\{p, q\})$ because $\{\neg p \wedge r\} \cup Th(\{p, q\})$ is inconsistent.

Let $\Pi = (\delta_0, \delta_1, \ldots)$ be a sequence of defaults in D without multiple occurrences.

—Π is a *constrained process* of the default theory $T = (W, D)$ iff, for all k such that $\Pi[k]$ is defined, δ_k is applicable to $(In(\Pi[k]), Con(\Pi[k]))$.

—A *closed constrained process* Π is a constrained process such that every default δ which is applicable to $(In(\Pi), Con(\Pi))$ already occurs in Π.

—A pair of sets of formulae (E, C) is a *constrained extension of* T iff there is a closed constrained process Π of T such that $(E, C) = (In(\Pi), Con(\Pi))$.

Note that we do not need a concept of success here because of the definition of default applicability we adopted: δ is only applicable to (E, C) if it does not lead to a contradiction. Let us reconsider the "broken arms" example: $T = (W, D)$ with $W = \{broken(a) \vee broken(b)\}$, and D consisting of the defaults

$$\delta_1 = \frac{true : usable(a) \wedge \neg broken(a)}{usable(a)},$$

$$\delta_2 = \frac{true : usable(b) \wedge \neg broken(b)}{usable(b)}.$$

It is easily seen that there are two closed constrained processes, (δ_1) and (δ_2), leading to two constrained extensions:

$$(Th(W \cup \{usable(a)\}), Th(\{broken(b), usable(a), \neg broken(a)\})),$$

and

$$(Th(W \cup \{usable(b)\}), Th(\{broken(a), usable(b), \neg broken(b)\}))$$

The effect of the definitions above is that it is impossible to apply both defaults together: after the application of, say, δ_1, $\neg broken(a)$ is included in the Con-set; together with $broken(a) \vee broken(b)$ it follows $broken(b)$, therefore δ_2 is blocked. The two alternative constrained extensions describe the two possible cases we would have intuitively expected.

For another example consider $T = (W, D)$ with $W = \{p\}$ and $D = \{p : \neg r/q, p : r/r\}$. T has two constrained extensions, $(Th(\{p, q\}), Th(\{p, q, \neg r\}))$ and $(Th(\{p, r\}), Th(\{p, r\}))$. Note that for both constrained extensions, the second component collects the assumptions supporting the first component.

5.2 A Fixpoint Characterization

Schaub's original definition of constrained extensions used a fixed-point equation [Schaub 1992]: Let $T = (W, D)$ be a default theory. For a set C of formulae let $\Theta_T(C)$ be the pair of smallest sets of formulae (E', C') such that

(1) $W \subseteq E' \subseteq C'$

(2) E' and C' are deductively closed

(3) For every $\varphi : \psi_1, \ldots, \psi_n/\chi \in D$, if $\varphi \in E'$ and $C \cup \{\psi_1, \ldots, \psi_n, \chi\}$ is consistent, then $\chi \in E'$ and $\{\psi_1, \ldots, \psi_n, \chi\} \subseteq C'$.

The following result shows that this definition is equivalent to the definition of constrained extensions from the previous subsection.

THEOREM 8. *(E, C) is a constrained extension of T iff $(E, C) = \Theta_T(C)$.*

THEOREM 9. *Every default theory has at least one constrained extension. Furthermore Constrained Default Logic is semi-monotonic.*

5.3 Interconnections

In the following we describe the relationship among the default logic variants presented so far.

THEOREM 10. *Let T be a default theory and $E = In(\Pi)$ an extension of T, where Π is a closed and successful process of T. If $E \cup \neg Out(\Pi)$ is consistent, then $(E, Th(E \cup \neg Out(\Pi)))$ is a constrained extension of T.*

The converse does not hold since the existence of an extension is not guaranteed. For example $T = (\emptyset, \{true : p/\neg p\})$ has the single constrained extension $(Th(\emptyset), Th(\emptyset))$, but no extension.

THEOREM 11. *Let T be a default theory and $E = In(\Pi)$ a modified extension of T, where Π is a maximal process of T. If $E \cup \neg Out(\Pi)$ is consistent, then $(E, Th(E \cup \neg Out(\Pi)))$ is a constrained extension of T.*

The example $T = (W, D)$ with $W = \emptyset$ and $D = \{true : p/q, true : \neg p/r\}$ shows that we cannot expect the first compo-

nent of a constrained extension to be a modified extension: T has the single modified extension $Th(\{q, r\})$, but possesses two constrained extensions, $(Th(\{q\}), Th(\{p, q\}))$ and $(Th(\{r\}), Th(\{\neg p, r\}))$. As the following result demonstrates, it is not accidental that for both constrained extensions, the first component is included in the modified extension.

THEOREM 12. *Let T be a default theory and (E, C) a constrained extension of T. Then there is a modified extension F of T such that $E \subseteq F$.*

The following examples illustrates well the difference between the three approaches. Consider the default theory $T = (W, D)$ with $W = \emptyset$ and

$$D = \left\{ \frac{true : p}{q}, \frac{true : \neg p}{r}, \frac{true : \neg q, \neg r}{s} \right\}.$$

T has the single extension

$$Th(\{q, r\}),$$

two modified extensions,

$$Th(\{q, r\})$$

$$Th(\{s\}),$$

and three constrained extensions

$$(Th(\{q\}), Th(\{q, p\}))$$

$$(Th(\{r\}), Th(\{r, \neg p\}))$$

$$(Th(\{s\}), Th(\{s, \neg q, \neg r\})).$$

This theory illustrates the essential differences of the three approaches discussed. Default Logic does not care about inconsistencies among justifications, and may run into inconsistencies. Thus the first two defaults can be applied together, while if the third default is applied first, then the process is not closed and subsequent application of another default leads to failure. Justified Default Logic avoids the latter situation, so we obtain an additional modified extension. Constrained Default Logic avoids running into failure, too, but additionally requires joint consistency of justifications; therefore the two first defaults cannot be applied in conjunction, as in the other two approaches. Thus we get three constrained extensions.

We conclude this section by noting that for normal default theories, all default logic approaches discussed are identical. In other words, they coincide for the "well-behaved" class of default theories, and seek to extend it in different directions.

THEOREM 13. *Let T be a normal default theory, and E a set of formulae. The following statements are equivalent.*

(a) *E is an extension of T.*

(b) *E is a modified extension of T.*

(c) *There exists a set of formulae C such that (E, C) is a constrained extension of T.*

6. PRIORITIES ON DEFAULTS

6.1 Motivation

Often it is desirable to have priorities among defaults indicating a preference on which default to apply in situations where several defaults are applicable. One approach is to use implicit criteria of preference such as specificity among defaults; systems in this category include Baader and Hollunder [1993]; Etherington and Reiter [1983]; Pearl [1990]; and Touretzky et al. [1987].

This family of approaches suffers from the problem that the criterion may not be sufficient to model different application domains. For example, while specificity is appropriate for taxonomic information found in biology, it cannot capture the legal principle that a more recent law is preferred to an older one.

Therefore other kinds of approaches may have to be used, in which the priority information is represented explicitly. In the following we discuss two such approaches, which are based on the idea of static priorities and dynamic priorities, respectively.

6.2 Static Priorities

In Prioritized Default Logic [Brewka 1994], the user *gives explicitly the priority order* in which defaults have to be applied in situations where more than one default is applicable. In the simplest case a strict well-ordering (that means, an irreflexive total order in which every subset of D has a smallest element) is used. Let us consider the default theory $T = (W, D)$ with $W = \{bird, penguin\}$ and D consisting of the defaults

$$\delta_1 = \frac{penguin : \neg flies}{\neg flies}, \delta_2 = \frac{bird : flies}{flies}.$$

Suppose a total order $\ll$ is given according to which δ_1 is preferred over δ_2, i.e. $\delta_1 \ll \delta_2$. Then T together with $\ll$ should only admit the extension $Th(\{bird, penguin, \neg flies\})$.

Let $T = (W, D)$ be a normal default theory, and $\ll$ a strict well order on D. A process $\Pi = (\delta_0, \delta_1, \ldots)$ of T is *generated by* $\ll$ iff, for all i, δ_i is the $\ll$ — minimal default from $\Pi - \Pi[i]$ that is applicable to $In(\Pi[i])$, provided that such a δ_i exists. Clearly, Π is closed; it is also successful since T is a normal default theory. We say that E is *generated by* $\ll$ iff $E = In(\Pi)$ for a process Π that is generated by $\ll$.

Π and E, as defined above, are uniquely determined. In the previous example, $\ll$ is defined by $\delta_1 \ll \delta_2$. (δ_1) is the process generated by $\ll$, and $Th(\{bird, penguin, \neg flies\})$ is the extension generated by $\ll$.

Of course, we cannot always expect the defaults to be ordered in a total way. Often defaults should be left incomparable with one another. In other words, requiring a total ordering of defaults will often be a kind of overspecification. Here is a simple example from politics.

According to conservative politicians, taxes should be cut without cutting spending; the latter is unnecessary because reduced taxes are supposed to lead to increased economic growth. Radical conservatives proclaim tax cuts and spending cuts because the government should be as lean as possible. According to social democrats, neither taxes nor spending should be cut because they believe government should afford welfare programs and create additional demand. This information can be expressed by the following defaults:

$$\delta_1 = \frac{conservative : taxCut \wedge \neg spendingCut}{taxCut \wedge \neg\, spendingCut}$$

$$\delta_2 = \frac{conservative \wedge radical : taxCut \wedge spendingCut}{taxCut \wedge spendingCut}$$

$$\delta_3 = \frac{socialDemocrat : \neg taxCut \wedge \neg spendingCut}{\neg taxCut \wedge \neg spendingCut}$$

Intuitively it is clear that δ_2 should be given higher priority than δ_1 because the information that somebody is a radical conservative is more specific than the information that she is a conservative. But there is no reason to prescribe any order between δ_1 and δ_3, or between δ_2 and δ_3. Thus we only have $\delta_2 < \delta_1$.

Technically speaking, the priority information will be given in the form of a *strict partial order* $<$ on the set of defaults. The definition of the semantics will make use of all strict well orders that contain $<$. This approach resembles the treatment of concurrency in computer science, where meaning is assigned by considering all possible linearizations.

Definition 2. $T = (W, D, <)$ is a *prioritized default theory* if (W, D) is a normal default theory and $<$ a strict partial order on D. E is a *PDL-extension of* T iff there is a strict well order $\ll$ on D which contains $<$ and generates E.

Consider the previous example, with the set of facts $W = \{conservative, radical\}$, and $<$ defined by $\delta_2 < \delta_1$. There are three strict well orders on D that contain $<$, namely

$$\delta_3 \ll \delta_2 \ll \delta_1,$$

$$\delta_2 \ll \delta_3 \ll \delta_1,$$

and

$$\delta_2 \ll \delta_1 \ll \delta_3.$$

It is easily seen that all of them lead to the same PDL-extension, namely $Th(\{conservative, radical, taxCut, spendingCut\})$.

By definition of a PDL-extension E, $E = In(\Pi)$ for a closed and successful process Π. Therefore we make the following observation:

THEOREM 14. *If E is a PDL-extension of the prioritized default theory $T = (W, D, <)$ then E is an extension of (W, D).*

The next result follows from the observation that given a finite set M, every strict partial order on M is included in a strict well order on M. A strict partial order $<$ defines some constraints about which elements should precede which other elements. A strict total order $\ll$ respecting these constraints can always be constructed. Since M is finite, a minimal element exists. Therefore the extension generated by $\ll$ is a PDL-extension.

THEOREM 15. *A prioritized default theory $T = (W, D, <)$ always has a PDL-extension if D is finite.*

The above theorem is not true for infinite sets of defaults. Let $D = \{\delta_i | i \geq 0\}$, and $<$ the strict partial order on D determined by $\delta_i < \delta_0$ for all $i > 0$. Then there is no strict well order of D containing $<$; there are, of course, strict total orders, but the existence of minimal elements cannot be established.

6.3 Dynamic Priorities

The main drawback of the previous approach is that the user has to provide all priority information, which is supposed to be static and not change over time. These requirements may be too strict for some problems. Brewka [1994] proposed an approach in which priority information is part of the logical language and can be derived like any other statement. For example, it is possible to write

$$\frac{p : d_1 < d_2}{d_1 < d_2}$$

to express that "In cases p is true, usually default d_1 should be given preference over default d_2". Of course there may be some exceptions in which $d_1 < d_2$ cannot be assumed due to other information being available.

As this example shows, the approach uses *names* to refer to defaults, and a preference relation $<$ as part of the language. Reasoning about priorities is very flexible, but the high expressiveness brings problems, too; for example, it is possible that even if all defaults are normal there is no extension. Consider the theory consisting of the following two defaults:

$$d_1 = \frac{true : d_2 < d_1}{d_2 < d_1},\ d_2 = \frac{true : d_1 < d_2}{d_1 < d_2}.$$

Default d_1 says that usually you should prefer default d_2 instead of d_1, and vice versa. Therefore this theory has no extension.

7. OTHER VARIANTS OF DEFAULT LOGIC

In this section we briefly review some further Default Logic approaches without going into technical details.

7.1 Rational Default Logic

Constrained Default Logic enforces joint consistency of the justifications of defaults that contribute to an extension, but goes one step further by requiring that the consequent of a default be consistent with the current Con-set. Rational Default Logic [Mikitiuk and Truszczynski 1995] does not require the latter step. Technically, a default $\varphi : \psi_1, \ldots, \psi_n / \chi$ is *rationally applicable* to a pair of deductively closed sets of formulae (E, C) iff $\varphi \in E$ and $\{\psi_1, \ldots, \psi_n\} \cup C$ is consistent.

As an example, consider the default theory $T = (W, D)$ with $W = \emptyset$ and $D = \{true : b/c, true : \neg b/d, true : \neg c/e, true : \neg d/f\}$. T has the single extension $Th(\{c, d\})$, three constrained extensions,

$$(Th(\{e, f\}), Th(\{e, \neg c, f, \neg d\}))$$
$$(Th(\{c, f\}), Th(\{c, b, f, \neg d\}))$$
$$(Th(\{d, e\}), Th(\{d, \neg b, e, \neg c\})) \quad (5)$$

but two rational extensions, $Th(\{c, f\})$ and $Th(\{d, e\})$. The first constrained extension is "lost" in Rational Default Logic because it is not closed under application of further defaults. $true : b/c$ and $true : \neg b/d$ are both rationally applicable to $Th(\{e, f\})$; but once one of them is applied to $Th(\{e, f\})$ we get a failed situation.

Mikitiuk and Truszczynski [1995] show that if E is an extension of T in Rational Default Logic, then (E, C) is a constrained extension of T for some set C. The converse is true for semi-normal default theories.

Rational Default Logic does not guarantee the existence of extensions. For example, the default theory consisting of the single default $true : p/\neg p$ does not have any extensions.

7.2 Cumulative Default Logic

As mentioned earlier, Cumulative Default Logic was introduced by Brewka to ensure the property of cumulativity [Brewka 1991]. The solution he adopted was to use so-called *assertions*, pairs (φ, J) of a formula φ and a set of formulae J, which collects the assumptions that were used to deduce φ. When a default is applied to deduce φ the justifications of that default are added to J.

We illustrate this approach by considering the example from subsection 3.3 which showed that Default Logic violates cumulativity. Consider $T = (W, D)$ with $W = \emptyset$ and D consisting of the defaults

$$\frac{true : a}{a}, \quad \frac{a \vee b : \neg a}{\neg a}.$$

In the beginning we can apply only the first default and derive the assertion $(a, \{a\})$, meaning that we derived a based on the assumption a. Obviously the second default is not applicable. The violation of cumulativity in Default Logic was caused by the addition of $a \vee b$ as a new fact which opened the way for the application of the second default instead of the first one. But in Cumulative Default Logic we are allowed to add the assertion $(a \vee b, \{a\})$ to the default theory (if a is derived based on a, then $a \vee b$ is also derived based on a), but now the second default is still not applicable because $\neg a$ is not consistent with the set of supporting beliefs $\{a\}$.

Note that adding a to the default the-

ory as we did in Default Logic corresponds to adding the assertion $(a, \emptyset)$, which is different from $(a, \{a\})$. If we disregard $\{a\}$, which is the assumption upon which the deduction of a was based, then indeed we can get more conclusions; this forgetting is the deeper reason for the failure of Default Logic to satisfy cumulativity.

From the technical and practical point of view, the use of assertions is complicated and causes practical problems, for example with regard to implementation; this is the price we have to pay for cumulativity. And the gain is questionable in the light of our discussion in subsection 3.3, which argued that lemmas can and should be represented as defaults, rather than facts. Nevertheless, Cumulative Default Logic was historically an important one.

7.3 Disjunctive Default Logic

The "broken arm" example from subsection 3.2 shows that Default Logic has a deficiency with the correct treatment of disjunctive information. Gelfond et al. [1991] proposes a way out of these difficulties by the following analysis: if a formula $\varphi \vee \psi$ becomes part of the current knowledge base (either as a fact or as a consequent of some default), it should *not* be included as a predicate logic formula. Instead, it should have the effect that one of φ and ψ becomes part of an extension. In other words, the expression

$$broken(a) | broken(b)$$

should have the effect that an extension contains one of the two disjuncts, rather than the disjunction $broken(a) \vee broken(b)$. To see another example, consider the default theory $T = (W, D)$ with $W = \{p \vee q\}$ and $D = \{p : r/r, q : r/r\}$. In Default Logic we know the formula $p \vee q$ but are unable to apply any of the two defaults; so we end up with the single extension $Th(\{p \vee q\})$. On the other hand, Disjunctive Default Logic leads to two extensions, one in which p is included, and one in which q is included. In the former case $p : r/r$ becomes applicable, in the latter case $q : r/r$ becomes applicable. So we end up with two extensions, $Th(\{p, r\})$ and $Th(\{q, r\})$, which is intuitively more appealing. For more details, see Gelfond et al. [1991].

7.4 Weak Extensions

All variants of Default Logic discussed so far share the same idea of treating prerequisites of defaults: in order for a default δ to be applicable, its prerequisite must be *proven using the facts and the consequents of defaults that were applied before* δ. For example, in order for the default $p : true/p$ to be applicable, p must follow from the facts, or be the consequent of another default etc. A default theory consisting only of this default has the single extension $Th(\emptyset)$.

This has led researchers to refer to Default Logic as being "strongly grounded" in the given facts. In contrast, Autoepistemic Logic [Moore 1985] provides more freedom in choosing what one wants to *believe* in. Weak extensions of default theories were introduced to capture this intuition in the default logic framework [Marek and Truszczynski 1993].

In the framework of weak extensions, we can simply decide to believe in some formulae. The only requirement is that this decision can be justified using the facts and default rules. Reconsider the default theory consisting of the single default $p : true/p$. We may decide to believe in p or not. Suppose we do believe in p; then the default can be applied and gives us p as a consequence. In this sense the default justifies the decision to believe in p; $Th(\{p\})$ is thus a weak extension. Of course, we could also adopt the more cautious view and decide not to believe in p; then the default is not applicable, so p cannot be

proved and our decision is again justified. In general, extensions of a theory T are also weak extensions of T. For a technical discussion, see Marek and Truszczynski [1993].

8. CONCLUSION

Default Logic is an important method of knowledge representation and reasoning, because it supports reasoning with incomplete information, and because defaults can be found naturally in many application domains, such as diagnostic problems, information retrieval, legal reasoning, regulations, specifications of systems and software, etc. Default Logic can be used either to model reasoning with incomplete information, which was the original motivation, or as a formalism which enables compact representation of information [Cadoli et al. 1994]. Important prerequisites for the development of successful applications in these domains include (i) the understanding of the basic concepts, and (ii) the existence of powerful implementations.

The aim of this paper is to contribute to the first requirement. We have discussed the basic concepts and ideas of Default Logic, and based the presentation on operational interpretations, rather than on fixpoints, as usually done. The operational interpretations allow learners to apply concepts to concrete problems in a straightforward way. This is an important point, because the difficulty of understanding Default Logic should not be underestimated. A survey among students at the University of Toronto regarding their ranking of the difficulty of several nonmonotonic reasoning formalisms resulted in Default Logic (presented based on fixpoints) being perceived as the second most difficult one, surpassed only by the full version of Circumscription [McCarthy 1980], but well ahead of Autoepistemic Logic [Moore 1985].

In this paper we have not discussed the semantics of Default Logic (see Besnard and Schaub [1994]; Engelfriet and Treur [1996]; Teng [1996]), or complexity issues (see Gottlob [1992]; Kautz and Selman [1989]). An argumentation-theoretic study of default logics is found in Bondarenko et al. [1997].

Implementation aspects were also outside the scope of this paper; some entry points to current work in the area include Cholewinski [1994]; Cholewinski et al. [1995]; Courtney et al. [1996]; Linke and Schaub [1995]; Niemela [1995]; Risch and Schwind [1994]; and Schaub [1995].

REFERENCES

ANTONIOU, G., O'NEILL, T., AND THURBON, J. 1996. Studying properties of classes of default logics: Preliminary report. In *Proceedings of the Fourth Pacific Rim International Conference on Artificial Intelligence*, Springer-Verlag, New York, 558–569.

ANTONIOU, G. AND SPERSCHNEIDER, V. 1994. Operational concepts of nonmonotonic logics. Part 1: Default logic. *Artif. Intell. Rev. 8*, 3–16.

BAADER, F. AND HOLLUNDER, B. 1993. How to prefer more specific defaults in terminological logics. In *Proceedings of the 13th International Joint Conference on Artificial Intelligence*, MIT Press, Cambridge, MA.

BESNARD, P. 1989. *An Introduction to Default Logic*. Springer-Verlag, Vienna, Austria.

BESNARD, P. AND SCHAUB, T. 1994. possible worlds semantics for default logics. *Fundam. Inf. 21*, 39–66.

BONDARENKO, A., DUNG, P. M., KOWALSKI, R. A., AND TONI, F. 1997. An abstract, argumentation-theoretic approach to default reasoning. *Artif. Intell. 93*, 1-2, 63–101.

BREWKA, G. 1991. Cumulative default logic: in defense of nonmonotonic inference rules. *Artif. Intell. 50*, 2 (July 1991), 183–205.

BREWKA, G. 1994. Reasoning about priorities in default logic. In *Proceedings of the 12th National Conference on Artificial Intelligence (vol. 2)* (AAAI'94, Seattle, WA, July 31–Aug. 4), B. Hayes-Roth and R. E. Korf, Eds. Amer. Assn. for Artificial Intelligence, Menlo Park, CA, 940–945.

CADOLI, M., DONINI, F. M., AND SCHAERF, M. 1994. Is intractability of non-monotonic reasoning a real drawback?. In *Proceedings of the 12th National Conference on Artificial Intelligence (vol. 2)* (AAAI'94, Seattle, WA, July 31–Aug. 4), B. Hayes-Roth and R. E. Korf, Eds. Amer. Assn. for Artificial Intelligence, Menlo Park, CA, 946–951.

CHOLEWINSKI, P. 1994. Stratified default logic. In *Proceedings of the Conference on Computer Science Logic*, Springer-Verlag, New York, 456–470.

CHOLEWINSKI, P., MAREK, W., MIKITIUK, A., AND TRUSZCZYNSKI, M. 1995. Experimenting

with default logic. In *Proceedings of the International Conference on Logic Programming*, MIT Press, Cambridge, MA.

COURTNEY, A., ANTONIOU, G., AND FOO, N. 1996. Exten: A system for computing default logic extensions. In *Proceedings of the Fourth Pacific Rim International Conference on Artificial Intelligence*, Springer-Verlag, New York, 471–482.

DELGRANDE, J. P., SCHAUB, T., AND JACKSON, W. K. 1994. Alternative approaches to default logic. *Artif. Intell. 70*, 1-2 (Oct. 1994), 167–237.

ENGELFRIET, J. AND TREUR, J. 1996. Semantics for default logic based on specific branching time models. In *Proceedings of the 12th European Conference on Artificial Intelligence*, John Wiley and Sons, Inc., New York, NY, 60–64.

ETHERINGTON, D W 1987. Formalizing nonmonotonic reasoning systems. *Artif. Intell. 31*, 1 (Jan. 1987), 41–85.

ETHERINGTON, D. 1987. *Reasoning with Incomplete Information*. Pitman Publishing, London, UK.

ETHERINGTON, D. AND REITER, R. 1983. On inheritance hierarchies with exceptions. In *Proceedings of the National Conference on Artificial Intelligence*, AAAI Press, Menlo Park, CA, 104–108.

GELFOND, M., LIFSCHITZ, V., PRZYMUSINSKA, H., AND TRUSZCZYNSKI, M. 1991. Disjunctive defaults. In *Proceedings of the 2nd International Conference on Principles of Knowledge Representation and Reasoning*, Morgan Kaufmann, San Mateo, CA, 230–237.

GOTTLOB, G. 1992. Complexity results for nonmonotonic logics. *J. Logic Comput. 2*, 397–425.

KAUTZ, H. A. AND SELMAN, B. 1989. Hard problems for simple default logics. In *Proceedings of the First International Conference on Principles of Knowledge Representation and Reasoning*, R. J. Brachman, H. J. Levesque, and R. Reiter, Eds. Morgan Kaufmann Publishers Inc., San Francisco, CA, 189–197.

LINKE, T AND SCHAUB, T. 1995. Lemma handling in default logic theorem proving. In *Proceedings of the Conference on Symbolic and Quantitative Approaches to Reasoning and Uncertainty*, Springer-Verlag, New York, 285–292.

LUKASZEWICZ, W. 1988. Considerations on default logic. *Comput. Intell. 4*, 1, 1–16.

LUKASZEWICZ, W. 1990. *Non-Monotonic Reasoning: Formalization of Commonsense Reasoning*. Ellis Horwood, Upper Saddle River, NJ.

MAKINSON, D. 1994. General patterns in nonmonotonic reasoning. In *Handbook of Logic in Artificial Intelligence and Logic Programming: Nonmonotonic Reasoning and Uncertain Reasoning (vol. 3)*, D. M. Gabbay, C. J. Hogger, and J. A. Robinson, Eds. Oxford University Press, Inc., New York, NY, 35–110.

MAREK, V. AND TRUSZCZYNSKI, M. 1993. *Nonmonotonic Reasoning*. Springer-Verlag, New York, NY.

MCCARTHY, J. 1980. Circumscription: A form of non-monotonic reasoning. *Artif. Intell. 13*, 27–39.

MIKITIUK, A. AND TRUSZCZYNSKI, M. 1995. Constrained and rational default logics. In *Proceedings of the 14th International Joint Conference on Artificial Intelligence*, Morgan Kaufmann, San Mateo, CA, 1509–1515.

MOORE, R. C. 1985. Semantical considerations on nonmonotonic logic. *Artif. Intell. 25*, 1 (Jan. 1985), 75–94.

NIEMELA, I. 1995. Towards efficient default reasoning. In *Proceedings of the 14th International Joint Conference on Artificial Intelligence*, Morgan Kaufmann, San Mateo, CA, 312–318.

PEARL, J. 1990. System z: A natural ordering of defaults with tractable applications to nonmonotonic reasoning. In *Proceedings of the Third International Conference on Theoretical Aspects of Reasoning about Knowledge*, Springer-Verlag, New York, NY.

POOLE, D. 1994. Default logic. In *Handbook of Logic in Artificial Intelligence and Logic Programming: Nonmonotonic Reasoning and Uncertain Reasoning (vol. 3)*, D. M. Gabbay, C. J. Hogger, and J. A. Robinson, Eds. Oxford University Press, Inc., New York, NY, 189–215.

REITER, R. 1978. On closed world databases. In *Logic and Data Bases*, H. Gallaire and J. Minker, Eds. Plenum Press, New York, NY, 55–76.

REITER, R. 1980. A logic for default reasoning. *Artif. Intell. 13*, 81–132.

RISCH, V. AND SCHWIND, C. 1994. Tableau-based characterization and theorem proving for default logic. *J. Autom. Reasoning 13*, 223–242.

SCHAUB, T. 1992. On constrained default theories. In *Proceedings of the 10th European Conference on Artificial Intelligence* (ECAI '92, Vienna, Austria, Aug. 3–7), B. Neumann, Ed. John Wiley and Sons, Inc., New York, NY, 304–308.

SCHAUB, T. 1995. A new methodology for query-answering in default logics via structure-oriented theorem proving. *J. Autom. Reasoning 15*, 1, 95–165.

TENG, C. 1996. Possible world partition sequences: A unifying framework for uncertain reasoning. In *Proceedings of 12th Conference on Uncertainty in Artificial Intelligence*, 517–524.

TOURETZKY, D., HORTY, J., AND THOMASON, R. 1987. A clash of intuitions: The current state of nonmonotonic multiple inheritance systems. In *Proceedings of the 10th International Joint Conference on Artificial Intelligence*, MIT Press, Cambridge, MA, 476–482.

Received: June 1996; revised: May 1997; accepted: August 1998