university of groningen

faculty of science and engineering

SYLLABUS DEFAULT LOGIC

Lecturer: Rineke Verbrugge

2020–2021

Adapted from

Nonmonotonic Reasoning

by Grigori Antoniou, with contributions by Mary-Anne Williams

MIT Press, Cambridge (MA), 1997

LaTeX version by Ramon Meffert

Last updated March 19, 2021 by Jelle Visser and Rineke Verbrugge

Part II

Default Logic

The first nonmonotonic logic we present is *Default Logic*. Introduced by Reiter in 1980, it has evolved as one of the main approaches to nonmonotonic reasoning and is (in the author's view) the most promising to be used for serious applications.

Default Logic distinguishes between two kinds of knowledge, usual predicate logic formulae (called *axioms* or *facts*) and rules of thumb called *defaults*. So, a default theory consists of a set of facts which represents certain, but usually incomplete information about the world; and a set of defaults that sanction plausible but not necessarily true conclusions. That means, some conclusions may have to be revised when more information becomes available. A simple example of a default is

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

which is read as 'If X is a bird and if it is consistent to assume that X flies, then conclude that X flies'. In the absence of knowledge to the contrary it is reasonable to assume that a particular bird can fly, more simply 'Usually (typically) birds fly'.

Given the information that Tweety[1] is a bird we may conclude that he flies. But if later on we learn that he cannot fly, for example because he is a penguin, then the default-becomes inapplicable: we are no longer allowed to assume that Tweety flies! Thus our behaviour is nonmonotonic. Chapter 3 introduces the notion of a default and gives an idea of possible application areas.

The operational semantics of Default Logic is defined in terms of so-called *extensions*, sets of beliefs one may hold about the domain described by the default theory under consideration. Their technical definition is complex and is presented in Chapter 4. Essentially, extensions are obtained by applying defaults as long as possible without running into inconsistencies.

One disappointing result is that some default theories may not possess any extensions; such theories are worthless because they don't provide any usable information. Therefore, various restricted classes of default theories have been analyzed.

[1]the most famous bird in the history of nonmonotonic reasoning!

Chapter 3

Default Reasoning

In section 3.1 we begin the presentation of Default Logic by introducing the notion of a default and presenting a variety of examples that demonstrate the broad applicational scope of the concept. In section 3.2 we introduce the syntax of Default Logic while in section 3.3 we informally discuss the semantics of the logic. The formal definition of the (operational) semantics will be given in the next chapter.

3.1 The notion of a default

Suppose I am asked in the morning how I intend to go to work. My answer is 'By bus' because usually I go to work by bus. This rule of thumb is represented by the default

$$\frac{goToWork : useBus}{useBus}$$

So I leave home, walk to the bus station and read a notice saying 'No buses today because we are on strike!'. Now I definitely know that I cannot use the bus to go to work, therefore the default is no longer applicable (now I cannot assume that I may use the bus). I have to revise my previous conclusion, so my behaviour is nonmonotonic.

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

$$goToWork \wedge \neg strike \rightarrow useBus.$$

Except that this rule is still insufficient: another reason for not wanting to use the bus is that I am late and therefore in a hurry. Then the rule could be modified as follows:

$$goToWork \wedge \neg strike \wedge \neg inHurry \rightarrow useBus.$$

But there are more conceivable reasons for my not using a bus, for example icy streets. A first difficulty of a predicate logic solution becomes apparent: we would have to list *all* possible obstacles, however unprobable they may be. Is this possible, especially in a rapidly changing environment?

But the real difficulty actually lies in the use of the rule: I would have to actively establish that all the preconditions of the rule are true (that is that the obstacles are not given) before

I could apply the rule. So every morning. I would have to call news agencies, listen to the radio etc. to obtain all the information needed about weather, strikes etc., before making up my mind. It might be afternoon when I would finally arrive at work, only to be told that I am fired because I missed giving my lecture.

Lt us now turn to more serious examples. Defaults can be used to model various forms of default reasoning. One such form is *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

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

To give another example: typically, if someone has a birthday, their friends give them gifts. Formulated as a default:

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

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:

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

Best-guess reasoning is exemplified by the following situation: I know there are some shopping centers in my city, and some of them are open on Sundays but I don't know which one is. On a Sunday I would drive to the closest one though I do not have any evidence that it will be open; it is simply the most convenient conjecture I can make:

$$\frac{closest(S) : openSundays(S)}{openSundays(S)}$$

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

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

in combination with the rule

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

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

$$cephalopod(X) \rightarrow mollusc(X) \wedge \neg shellBearer(X).$$

In *diagnosis*, the task is to determine some explanation for the symptoms observed; diagnosis is common in areas like medicine or factory quality control. Here is a funny (silly, if you wish) example from an everyday situation: if the car cannot start and if it was inspected recently, then it is out of gas. In default form:

$$\frac{\neg starts(X) \wedge recentlyInspected(X) : outOfGas(X)}{outOfGas(X)}$$

Defaults can be used naturally to model situations that are well-known in some fields of Computer Science and Artificial Intelligence. One such application is the *Closed World Assumption* 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 non-parameterized statement about single objects) is taken to be false in the problem domain if it does not follow from the axioms. To give an example, suppose you are consulting a database of scheduled flights. If a flight from London to Paris at 2am is not listed then it is natural to assume that it does not exist (even if this is wrong in case the database is incomplete). 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 $\neg phi$.

Finally we take a look at *reasoning about action*. Imagine a robot acting in an environment. For example, it may pick up a box and place it somewhere else. When actions are performed, then some properties change but most properties remain unaffected. For instance, if the robot picks up block A and puts it on the table, then the formula $on(A, B)$ is no longer correct. On the other hand, neither the colour of the blocks nor the location of blocks B and C have been changed. To represent the invariant aspects of the world *explicitly* would require a huge number of axioms. This problem is known as the *frame problem*. Using defaults it is very easy to express: 'All aspects of the world remain unchanged except for those that are explicitly changed by the action'.

$$\frac{holds(\varphi, S) : holds(\varphi, act(S))}{holds(\varphi, act(S))}$$

for all formulae φ, situations S and actions *act*. If φ is affected by *act* then $holds(\varphi, act(S))$ will be *explicitly known* to be false (as a postcondition of the action), so the default will be blocked[1].

[1]Unfortunately, this representation does not *automatically* give a satisfactory solution to the frame problem, but that is another story.

When a robot decides to take an action the question arises whether the action was carried out successfully. This is known as the *qualification problem.* Success of an action may depend on numerous factors. For instance, the success of an action 'Put block A on top of block B' depends on factors like A not being too heavy, the robot's arm not being broken, block A not being fluid etc. Again, the problem is naturally and simply represented using defaults.

3.2 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 φ, $\psi_1, \ldots,$ ψ_n, χ 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.

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

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

is not a default according to the definition above. Let us call such 'defaults' *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 φ, $\psi_1, \ldots,$ ψ_n, χ 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 occurring 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

$$\left(\{bird(tweety), bird(sam)\}, \left\{\frac{bird(tweety) : flies(tweety)}{flies(tweety)}, \frac{bird(sam) : flies(sam)}{flies(sam)}\right\}\right)$$

In order to explain why we interpret open defaults as schemata let us briefly show that two straightforward alternative interpretations are inadequate. In the first one we interpret open formulae occurring in a default as in predicate logic, that means as being universally quantified. Consider the open default

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

According to the interpretation proposed, the default would read as follows: *If all X are birds, and if for all X we may assume that they fly, then we conclude that all X fly.* Obviously this interpretation does not match our intuitive understanding. In fact, it would be useless: it could be applied only in the case that every object in the domain is a bird, and if no non-flying bird is known!

So let us interpret the formulae existentially. The same default would read as: *If there is a bird, and if there is an X that flies, then conclude that there is some flying object.* Using this interpretation, it would be impossible to conclude $\mathit{flies}(\mathit{tweety})$ from $\mathit{bird}(\mathit{tweety})$; instead, we could only conclude $\exists X \mathit{flies}(X)$.

We conclude the discussion of this point by noting that there is work being done to assign other interpretations to open defaults, but no satisfactory solution has been found to date.

3.3 Informal discussions of semantics

Given a default

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

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 where φ should be included 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

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

which says 'Usually my friends' friends are also my friends'. Given the information $\mathit{friend}(\mathit{tom}, \mathit{bob})$, $\mathit{friend}(\mathit{bob}, \mathit{sally})$, $\mathit{friend}(\mathit{sally}, \mathit{tina})$, we would like to conclude $\mathit{friend}(\mathit{tom}, \mathit{tina})$. But this is only possible if we apply the appropriate instance of the default schema to $\mathit{friend}(\mathit{sally}, \mathit{tina})$ and $\mathit{friend}(\mathit{tom}, \mathit{sally})$; The latter formula stems from a previous application of the default schema[2]. If we did not admit this intermediate step and used the original facts only, then we could not get the expected result.

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

$$\delta_1 = \frac{\mathit{green} : \neg \mathit{likesCars}}{\neg \mathit{likesCars}}, \quad \delta_2 = \frac{\mathit{aaaMember} : \mathit{likesCars}}{\mathit{likesCars}}$$

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 $\mathit{likesCars}$ and $\neg \mathit{likesCars}$ which is a contradiction. It is unintuitive to let the application of *rules of thumb* 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 $\neg \mathit{likesCars}$, so it is not consistent to assume $\mathit{likesCars}$. After these examples let us give a somewhat more precise formulation of default interpretation.

[2] with other instantiations, of course.

> *If φ is currently known, and if all ψ_i are consistent with the current knowledge base, then conclude χ. The current knowledge base E is obtained from the facts and the consequents of some defaults that have been applied previously.*

Here is the formal definition:

> $\delta = \frac{\varphi:\psi_1,\ldots,\psi_n}{\chi}$ 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 each other. 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 chapter. Here we just collect some desirable properties of extensions.

- An extension should include the set W of facts since W contains the certain information.
- An extension 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.
- An extension E should be *closed under the application of defaults in D* (formally: if $\frac{\varphi:\psi_1,\ldots,\psi_n}{\chi} \in D$, $\varphi \in E$ 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 $D = \{\frac{aussie:drinksBeer}{drinksBeer}\}$. Let $E = Th(\{aussie, \neg drinksBeer\})$. It is easy to check 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'!

We conclude the discussion of extensions with an example which shows that finding the appropriate formal definition is not simple, This should not scare the reader, of course, because they must 'only' understand the definition given in Chapter 4; it was Reiter who did the job for them! Now to our example: imagine a bank's credit approval procedure. Among others, it could be based on the defaults

$$\frac{true : creditworthy}{approveCredit},\ \frac{true : \neg creditworthy}{\neg creditworthy}.$$

The defaults should have the effect that if an applicant may be assumed to be creditworthy then credit is approved. But the bank is cautious in that it assumes by default that somebody is not creditworthy unless evidence to the contrary is provided. So, if no further information apart from both defaults is given, we would expect that credit is not approved.

Now suppose that the first default is applied; this may be done since we have no knowledge that *creditworthy* may not be assumed. Next we also apply the second default; this is possible since $\neg$*creditworthy* is consistent with the current knowledge. As a consequence, its consequent $\neg$*creditworthy* is included in the current knowledge base. We do not have an inconsistency, and yet something has gone wrong: inclusion of $\neg$*creditworthy* in the knowledge shows *a posteriori* that we should not have assumed *creditworthy* as done when the first default was applied. According to the definition given in the next chapter, the current knowledge base $Th(\{approveCredit, \neg creditworthy\})$ is *not* an extension, and this is in accordance with our intuitive expectation.

Problems

3-1. In the Aussie/Beer example in section 3.3, verify that $E = Th(\{aussie, \neg drinksBeer\})$ is a minimal set of formulae which is deductively closed, includes the default theory's facts, and is closed under the application of defaults. Can you think of an additional condition that would prevent E from being a candidate extension?

3-2.* Think of a problem domain other than those described here, and formulate some defaults within that domain.

Chapter 4

Operational Semantics of Default Logic

After having informally discussed the motivations and intuitions of Default Logic, in this chapter we present its formal operational semantics and derive some properties. The main focus is to provide a formal definition of the notion of an extension.

The definition we give in section 4.1 is operational in the sense that it is a procedure that can be applied to examples. The main idea consists of applying defaults as long as possible. If we find out that a default should not have been applied, then we have to backtrack and try some alternative. It is possible to give very simple prototype implementations of this procedure, and we do so in section 4.3 where we present a simple Prolog program that computes extensions. In section 4.4 we give an alternative characterization of extensions based on fixed-points; this characterization is the definition usually found in the literature. Finally, section 4.5 contains some theoretical results on Default Logic.

4.1 The definition of extensions

For a given default theory $T = (W, D)$, let $\Pi = (\delta_0, \delta_1, \dots)$ 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 by $\Pi[k]$ 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 $Th(M)$, where $M = W \cup \{cons(\delta) \mid \delta \text{ occurs in } \Pi\}$. 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.

- $Out(\Pi) = \{\neg\psi \mid \psi \in just(\delta) \text{ for some } \delta \text{ occurring in } \Pi\}$. 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:

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

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

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. 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 – the reader is asked to explain why $\Pi' = (\delta_1, \delta_2)$ is not a process. 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*. Success captures the intuitive statement 'nothing has gone wrong' in the sense that it was okay to have assumed the justifications of the defaults that have been applied to be true: no formula $\neg\psi$ in the *Out*-set has become part of the current knowledge base, so it was consistent to assume ψ.

- 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.

The reader may already have the feeling that closed and successful processes will play an important role in the definition of extensions because they exhibit some of the ideas we informally developed in the previous chapter. But before we proceed let us look at an example. Consider the default theory $T = (W, D)$ with $W = \{a\}$ and D containing the following defaults:

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

$\Pi_1 = (\delta_1)$ is successful but not closed since δ_2 may be applied to $In(\Pi_1) = Th(\{a, d\})$. $\Pi_1 = (\delta_1, \delta_2)$ is closed but not successful: both $In(\Pi_2) = Th(\{a, b, d\})$ 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, $In(\Pi_3) = Th(\{a, b\})$ is an extension of T, in fact its single extension.

Definition 4.1 (Extension)**.** *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 $E = In(\Pi)$.*

The definition of extensions is operational in that it may be directly applied to concrete examples. Nevertheless we make some remarks and give some tips for determining extensions.

No care needs to be taken to ensure the success of a process. If, by applying defaults in a particular order, we reach a failed situation (nonempty intersection of *In*-set and *Out*-set), then we simply backtrack along the process we have just built up and try some other alternative.

In the case of finite default theories (theories with a finite set D of defaults) there is no problem ensuring closure: simply apply any applicable default (which haven't been applied yet) until no more are left. But in the case where the set D is infinite we should be more careful in order not to deliberately avoid some default. Let us formulate and prove a simple result.

Lemma 4.2. *An infinite process* Π *of a default theory* $T = (W, D)$ *is closed iff each default that is applicable to* $In(\Pi[k])$*, for infinitely many numbers* k*, is already contained in* Π.

Proof:* By definition of processes and using the Compactness Theorem of predicate logic, it is easy to see that the following statements are equivalent for each formula ψ (to be thought of as justification of a default):

1. ψ is consistent with $In(\Pi)$.
2. ψ is consistent with $In(\Pi[k])$ for infinitely many k.
3. ψ is consistent with $In(\Pi[k])$ for all $k > k'$ (for some number k').

Likewise, the following statements are equivalent for each formula φ (to be thought of as the prerequisite of a default):

1. $\varphi \in In(\Pi)$
2. $\varphi \in In(\Pi[k])$ for infinitely many k.
3. $\varphi \in In(\Pi[k])$ for all $k > k'$ (for some number k').

These equivalences obviously prove the claim of the lemma. ■

So, a strategy to guarantee the closure of an infinite process Π must take care that any default which, from some stage k on, constantly demands application will eventually be applied. This is nothing other than *fairness*, a well-known concept in the area of concurrent programming.

We conclude this section by describing a formal way of constructing a systematic overview of all closed and successful processes. One reason for doing so is that a common error when trying to determine the extensions of a default theory is that some extensions are overlooked.

We intend to arrange all possible processes in a canonical manner within a tree, called the process tree of the given default theory $T = (W, D)$. 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.

The root of the process tree is labeled with $Th(W)$ as *In*-set and $\emptyset$ as *Out*-set (it should be clear why we use these labels before any default has been applied yet; they correspond to $In(())$ and $Out(())$ where $()$ is the empty sequence of defaults).

Now consider a node N labeled with sets *In* and *Out*. N is only expanded if $In \cap Out = \emptyset$ else it is marked as failed and is a leaf of the process tree. If N is expanded it possesses one successor node $N(\delta)$ for each default $\delta = \frac{\varphi:\psi_1,\ldots,\psi_n}{\chi}$ that does not appear on the path from the root to N and that is applicable to *In*. The edge from N to $N(\delta)$ is labeled with δ, and the labels of $N(\delta)$ are $Th(In \cup \{\chi\})$ and $Out \cup \{\neg\psi_1, \ldots, \neg\psi_n\}$.

In case a node N is allowed to be expanded but there are no applicable defaults left, N is marked as closed and successful. Of course, the *In*-set associated with N is an extension of the default theory.

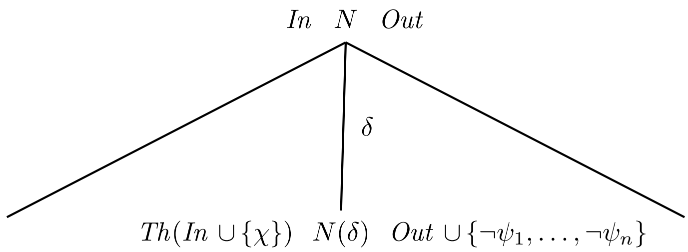

Figure 4.1
Expanding a node in a process tree

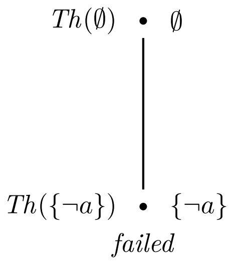

Figure 4.2

4.2 Some examples

Let $T = (W, D)$ with $W = \emptyset$ and $D = \{\frac{true:a}{\neg a}\}$. The process tree in Figure 4.2 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 knowledge base, so the default invalidates its own application because both the *In* and the *Out*-set contain $\neg a$. The reader hopefully agrees that this default exhibits a very strange behaviour. The example demonstrates that there need not always be an extension of a default theory.

Let $T = (W, D)$ with $W = \{\neg p, q\}$ and $D = \{\frac{q:\neg r}{p}\}$. Figure 4.3 shows that T has no extension. The default can be applied, but leads to a failed process: The *In*-set is inconsistent, therefore it includes r.

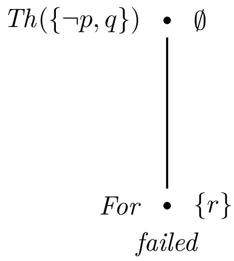

Figure 4.3

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

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

The process tree of T is found in Figure 4.4 and shows that T has exactly one extension, namely $Th(\{\neg q\})$. The right path of the tree gives us an example where application of a default destroys the applicability of a previous default. Reconsider the example of Greens who are members of AAA from Chapter 3.

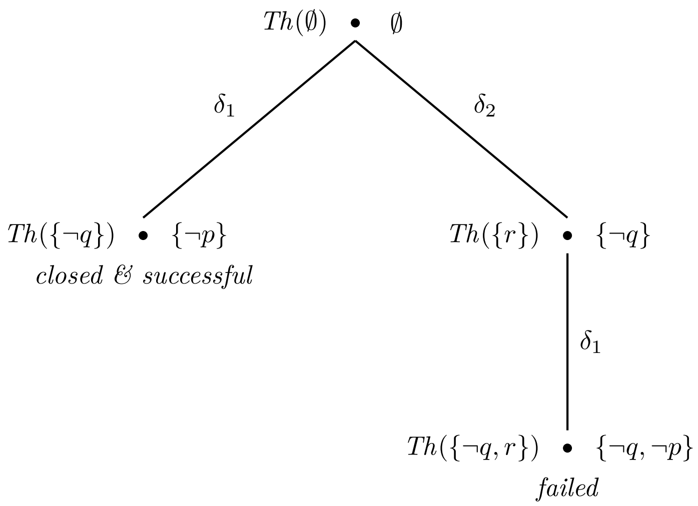

Figure 4.4

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

$$\delta_1 = \frac{green : \neg likesCars}{\neg likesCars}, \quad \delta_2 = \frac{aaaMember : likesCars}{likesCars}$$

The process tree in Figure 4.5 shows that T has exactly two extensions. This example shows a situation where two defaults contradict one another. A consequence is that there are two extensions which, taken together, are inconsistent. In fact, one of the strengths of Default Logic (and Nonmonotonic Reasoning in general) is that potentially inconsistent information may be represented within the same knowledge base.

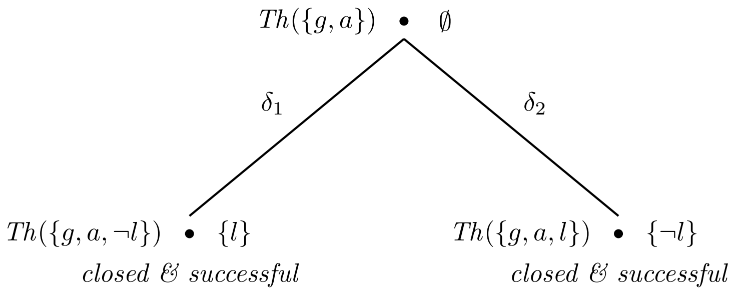

Figure 4.5

The examples discussed here show that a default theory may possess none, one or several extensions. We hope to have illustrated the basic concepts of Default Logic in a satisfactory way. From now on, we shall not draw process trees when discussing examples. The reader should consider it as a constant exercise to check the examples in subsequent sections and chapters by hand, in mind or using the prototype program of the following section. Who knows, maybe they will be rewarded by finding an error in one of the examples we will be discussing!