CHAPTER

Default Reasoning 12

Commonsense reasoning requires default reasoning. As we engage in commonsense reasoning, we make certain assumptions so that we can proceed. If we later gain more information, then we may have to revise those assumptions. For example, if we are told that Nathan went to sleep, we might assume that he went to sleep in his home. If we are told that he cooked breakfast, we might imagine him in his kitchen. If we later learn that he was actually at a hotel, our understanding of these events will have to be modified.

We have so far considered two special cases of default reasoning: assuming that unexpected events do not occur and assuming that events do not have unexpected effects. In this chapter, we consider default reasoning in general. We discuss atemporal default reasoning and then temporal default reasoning. We describe a general method for default reasoning about time based on the circumscription of abnormality predicates and how this method can be used to address the qualification problem. We then discuss the representation of default events and properties.

12.1 ATEMPORAL DEFAULT REASONING

We start by introducing default reasoning with some examples that do not involve time. Consider the following formula:

$$Apple(x) \Rightarrow Red(x)$$

This states that apples are always red. If we wish to represent that apples are sometimes red, we may use a formula such as the following:

$$Apple(x) \land \neg Ab_1(x) \Rightarrow Red(x) \tag{12.1}$$

The predicate $Ab_1(x)$ is called an *abnormality predicate*. It represents that x is abnormal in a way that prevents us from concluding that it is red given that it is an apple. Thus, (12.1) states that all apples that are not abnormal in fashion 1 are red. How do we specify when an apple is abnormal in fashion 1?

One possibility is that we could specify the necessary and sufficient conditions. We could represent that an apple is abnormal in fashion 1 if and only if it is a Granny Smith apple or it is rotten:

$$Ab_1(x) \Leftrightarrow GrannySmith(x) \lor Rotten(x) \tag{12.2}$$

But what about other conditions that might make an apple abnormal in fashion 1? If we wished to add other conditions, then we would have to modify (12.2):

$$\begin{aligned} &Ab_1(x) \Leftrightarrow \\ &GrannySmith(x) \vee Rotten(x) \vee NewtownPippin(x) \end{aligned} \tag{12.3}$$

An alternative is to use the following approach, which is more elaboration tolerant. We start by specifying some sufficient conditions for an apple being abnormal in fashion 1:

$$GrannySmith(x) \Rightarrow Ab_1(x) \tag{12.4}$$

$$Rotten(x) \Rightarrow Ab_1(x) \tag{12.5}$$

$$NewtownPippin(x) \Rightarrow Ab_1(x) \tag{12.6}$$

Then we circumscribe Ab_1 in formulas (12.4) to (12.6). Let Θ be the conjunction of (12.4), (12.5), and (12.6). By Theorem 2.1, $CIRC[\Theta; Ab_1]$ is equivalent to (12.3). This circumscription allows us to jump to the conclusion that the necessary and sufficient conditions are specified by (12.3), given that the sufficient conditions are specified by Θ.

Now suppose that a particular apple A is not a Granny Smith apple, not rotten, and not a Newtown Pippin apple:

$$\begin{gathered} Apple(A) \\ \neg GrannySmith(A) \\ \neg Rotten(A) \\ \neg NewtownPippin(A) \end{gathered}$$

We can conclude from (12.3) that A is not abnormal in fashion 1:

$$\neg Ab_1(A)$$

From this and (12.1), we can conclude that A is red:

$$Red(A)$$

We jump to the conclusion that A is red given that it is not a Granny Smith apple, not rotten, and not a Newtown Pippin apple.

12.2 TEMPORAL DEFAULT REASONING

Now we introduce time. In fact, previous chapters have already discussed a form of default reasoning about time. The event calculus assumes by default that (1) unexpected events do not occur and (2) events have no unexpected effects.

12.2.1 EVENT OCCURRENCES

Although it is abnormal for an event to occur at any given timepoint, events sometimes do occur. We specify what events are known to occur in the conjunction Δ of a domain description, which is described in Section 2.7. We write event calculus formulas containing the predicate $Happens(e, t)$, which represents that timepoint t is abnormal with respect to event e. We then use circumscription to minimize the extension of *Happens* in the conjunction Δ, which minimizes event occurrences.

12.2.2 EVENT EFFECTS

It is also abnormal for an event to have any given effect. We specify the known effects of events in the conjunction Σ of a domain description. We write event calculus formulas containing the abnormality predicates *Initiates*, *Terminates*, and *Releases*. We then circumscribe these predicates in the conjunction Σ in order to minimize event effects.

12.2.3 USING MINIMIZED EVENTS AND EFFECTS

Having minimized *Happens*, *Initiates*, *Terminates*, and *Releases*, we then write formulas containing these predicates that are outside the scope of the circumscription of these predicates. An example is axiom DEC5 of the discrete event calculus:

$$\begin{array}{c} \mathit{HoldsAt}(f,t) \land \neg \mathit{ReleasedAt}(f,t+1) \land \\ \neg \exists e\,(\mathit{Happens}(e,t) \land \mathit{Terminates}(e,f,t)) \Rightarrow \\ \mathit{HoldsAt}(f,t+1) \end{array}$$

This axiom states that a fluent f that is true at timepoint t is also true at $t+1$, unless f is released from the commonsense law of inertia at $t+1$ or there is an event e that occurs at t and terminates f at t.

12.3 DEFAULT REASONING METHOD

We now present the following general method for default reasoning about time based on the method just discussed:

1. We specify a set of abnormality predicates. Typically we use predicate symbols of the form Ab_i, where i is a positive integer.
2. We use the abnormality predicates within various event calculus formulas such as effect axioms. (Table 2.1 provides a list of types of event calculus formulas.)
3. We form a conjunction Θ of cancellation axioms. If γ is a condition, ρ is an abnormality predicate symbol of arity $n+1$, $\tau_1, \ldots, \tau_n$ are terms, and τ is a timepoint term, then

   $$\gamma \Rightarrow \rho(\tau_1, \ldots, \tau_n, \tau)$$

   is a *cancellation axiom* for ρ.
4. We compute the circumscription of the abnormality predicates in the conjunction of cancellation axioms Θ.

This method requires that we add the following clause to the definition of a condition in Section 2.5: If ρ is an abnormality predicate symbol of arity $n+1$, $\tau_1, \ldots, \tau_n$ are terms, and τ is a timepoint term, then $\rho(\tau_1, \ldots, \tau_n, \tau)$ is a condition.

The domain description is

$$CIRC[\Sigma; Initiates, Terminates, Releases] \wedge CIRC[\Delta_1 \wedge \Delta_2; Happens] \wedge$$
$$CIRC[\Theta; \rho_1, \ldots, \rho_n] \wedge \Omega \wedge \Psi \wedge \Pi \wedge \Gamma \wedge \mathrm{E} \wedge \mathrm{CC}$$

where $\rho_1, \ldots, \rho_n$ are the abnormality predicate symbols, Θ is a conjunction of cancellation axioms containing $\rho_1, \ldots, \rho_n$, and Σ, Δ_1, Δ_2, Ω, Ψ, Π, Γ, E, and CC are as defined in Section 2.7.

12.4 DEFAULTS AND THE QUALIFICATION PROBLEM

Default reasoning can be used to address the qualification problem (introduced in Section 3.3). Default reasoning provides a more elaboration-tolerant way of specifying qualifications that prevent events from having their intended effects.

First, we add abnormality predicates to effect axioms, which are the same as fluent precondition axioms. We write effect axioms of the form

$$\gamma \wedge \neg\rho(\tau_1, \ldots, \tau_n, \tau) \Rightarrow Initiates(\alpha, \beta, \tau)$$

or

$$\gamma \wedge \neg\rho(\tau_1, \ldots, \tau_n, \tau) \Rightarrow Terminates(\alpha, \beta, \tau)$$

where γ is a condition, ρ is an abnormality predicate symbol of arity $n + 1$, $\tau_1, \ldots, \tau_n$ are terms, τ is a timepoint term, α is an event term, and β is a fluent term. Second, whenever we wish to add a qualification, we add a cancellation axiom for ρ.

Default reasoning can also be used to represent qualifications that prevent events from occurring. But action precondition axioms (discussed in Section 3.3.2) already provide an elaboration tolerant way of expressing such qualifications.

12.4.1 EXAMPLE: DEVICE REVISITED

Consider the example of turning on a device. We introduce the abnormality predicate $Ab_1(d, t)$, which represents that device d is abnormal in some way at timepoint t. We use this predicate in an effect axiom. If a device is not Ab_1 and an agent turns on the device, then the device will be on:

$$\neg Ab_1(d, t) \Rightarrow Initiates(TurnOn(a, d), On(d), t) \tag{12.7}$$

We add cancellation axioms that represent qualifications. A device is Ab_1 if its switch is broken:

$$HoldsAt(BrokenSwitch(d), t) \Rightarrow Ab_1(d, t) \tag{12.8}$$

A device is Ab_1 if it is not plugged in:

$$\neg HoldsAt(PluggedIn(d), t) \Rightarrow Ab_1(d, t) \tag{12.9}$$

A particular antique device never works:

$$Ab_1(AntiqueDevice1, t) \tag{12.10}$$

Now consider the following observations and narrative in which the device is initially off and then the device is turned on:

$$\neg HoldsAt(On(Device1), 0) \tag{12.11}$$

$$\neg ReleasedAt(f, t) \tag{12.12}$$

$$Happens(TurnOn(Nathan, Device1), 0) \tag{12.13}$$

Suppose that the device's switch is not broken, the device is plugged in, and the device is not the antique device:

$$\neg HoldsAt(BrokenSwitch(Device1), 0) \tag{12.14}$$

$$HoldsAt(PluggedIn(Device1), 0) \tag{12.15}$$

$$Device1 \neq AntiqueDevice1 \tag{12.16}$$

We can then show that the device will be on at timepoint 1.

Proposition 12.1. *Let* $\Sigma = (12.7)$, $\Delta = (12.13)$, $\Theta = (12.8) \wedge (12.9) \wedge (12.10)$, $\Omega = (12.16) \wedge U[On, BrokenSwitch, PluggedIn]$, *and* $\Gamma = (12.11) \wedge (12.12) \wedge (12.14) \wedge (12.15)$. *Then we have*

$$\begin{aligned} &CIRC[\Sigma; Initiates, Terminates, Releases] \wedge CIRC[\Delta; Happens] \wedge \\ &\qquad CIRC[\Theta; Ab_1] \wedge \Omega \wedge \Gamma \wedge \mathrm{DEC} \\ &\qquad \vdash HoldsAt(On(Device1), 1) \end{aligned}$$

Proof. From $CIRC[\Sigma; Initiates, Terminates, Releases]$ and Theorems 2.1 and 2.2, we have

$$\begin{aligned} &Initiates(e, f, t) \Leftrightarrow \\ &\exists a, d\ (e = TurnOn(a, d) \wedge f = On(d) \wedge \neg Ab_1(d, t)) \end{aligned} \tag{12.17}$$

$$\neg Terminates(e, f, t) \tag{12.18}$$

$$\neg Releases(e, f, t) \tag{12.19}$$

From $CIRC[\Delta; Happens]$ and Theorem 2.1, we have

$$Happens(e, t) \Leftrightarrow (e = TurnOn(Nathan, Device1) \wedge t = 0) \tag{12.20}$$

From $CIRC[\Theta; Ab_1]$ and Theorem 2.1, we have

$$\begin{aligned} &Ab_1(d, t) \Leftrightarrow \\ &HoldsAt(BrokenSwitch(d), t) \vee \\ &\neg HoldsAt(PluggedIn(d), t) \vee \\ &d = AntiqueDevice1 \end{aligned} \tag{12.21}$$

From this, (12.14), (12.15), and (12.16), we have $\neg Ab_1(Device1, 0)$. From this, (12.13) (which follows from (12.20)), (12.7) (which follows from (12.17)), and DEC9, we have $HoldsAt(On(Device1), 1)$. ■

12.4.2 EXAMPLE: BROKEN DEVICE

Suppose that instead of (12.14), (12.15), and (12.16), we have the fact that the device's switch is broken:

$$HoldsAt(BrokenSwitch(Device1), 0) \tag{12.22}$$

We can then show that the device will *not* be on at timepoint 1.

Proposition 12.2. *Let* $\Sigma = (12.7)$, $\Delta = (12.13)$, $\Theta = (12.8) \wedge (12.9) \wedge (12.10)$, $\Omega = U[On, BrokenSwitch, PluggedIn]$, *and* $\Gamma = (12.11) \wedge (12.12) \wedge (12.22)$. *Then we have*

$$\begin{aligned} &CIRC[\Sigma; Initiates, Terminates, Releases] \wedge CIRC[\Delta; Happens] \wedge \\ &\qquad CIRC[\Theta; Ab_1] \wedge \Omega \wedge \Gamma \wedge \mathrm{DEC} \\ &\qquad \vdash \neg HoldsAt(On(Device1), 1) \end{aligned}$$

Proof. From $CIRC[\Sigma; Initiates, Terminates, Releases]$ and Theorems 2.1 and 2.2, we have (12.17), (12.18), and (12.19). From $CIRC[\Delta; Happens]$ and Theorem 2.1, we have (12.20). From $CIRC[\Theta; Ab_1]$ and Theorem 2.1, we have (12.21). From (12.21) and (12.22), we have $Ab_1(Device1, 0)$. From this and (12.17), we have $\neg\exists e\,(Happens(e, 0) \wedge Initiates(e, On(Device1), 0))$. From this, (12.11), (12.12), and DEC6, we have $\neg HoldsAt(On(Device1), 1)$. ■

12.4.3 STRONG AND WEAK QUALIFICATIONS

Patrick Doherty and Jonas Kvarnström proposed a distinction between strong and weak qualifications. If an event is *strongly qualified*, then it will not have its intended effects; if an event is *weakly qualified*, then it may or may not have its intended effects. A weakly qualified event results in two classes of models: one in which the event has its intended effects and one in which the event does not. Thus, weakly qualified events have nondeterministic effects. Weak qualifications can be represented using determining fluents (introduced in Section 9.1). A determining fluent is introduced for each weak qualification.

12.4.4 EXAMPLE: ERRATIC DEVICE

Let us extend our device example to erratic devices. If a device is erratic, then when it is turned on, it may or may not go on. We introduce the determining fluent $DeterminingFluent(d)$. We add a cancellation axiom that states that if a device is erratic and the determining fluent is true, then the device is Ab_1:

$$\begin{gathered} HoldsAt(Erratic(d), t) \wedge \\ HoldsAt(DeterminingFluent(d), t) \Rightarrow \\ Ab_1(d, t) \end{gathered} \tag{12.23}$$

Now suppose that the device's switch is not broken, the device is plugged in, the device is erratic, and Nathan turns on the device at timepoint 0. We can show that if $DeterminingFluent(Device1)$ is false at timepoint 0, then the device will be on at timepoint 1. But if $DeterminingFluent(Device1)$ is true at timepoint 0, then the device will not be on at timepoint 1. We use the following observations. The device is erratic:

$$HoldsAt(Erratic(Device1), 0) \tag{12.24}$$

The determining fluent is always released from the commonsense law of inertia, and *On*, *BrokenSwitch*, *PluggedIn*, and *Erratic* are never released from this law:

$$ReleasedAt(DeterminingFluent(d), t) \tag{12.25}$$

$$\neg ReleasedAt(On(d), t) \tag{12.26}$$

$$\neg ReleasedAt(BrokenSwitch(d), t) \tag{12.27}$$

$$\neg ReleasedAt(PluggedIn(d), t) \tag{12.28}$$

$$\neg ReleasedAt(Erratic(d), t) \tag{12.29}$$

Proposition 12.3. *Let* $\Sigma = (12.7)$, $\Delta = (12.13)$, $\Theta = (12.8) \wedge (12.9) \wedge (12.23)$, $\Omega = U[On, BrokenSwitch, PluggedIn, Erratic, DeterminingFluent]$, *and* $\Gamma = (12.11) \wedge (12.14) \wedge (12.15) \wedge (12.24) \wedge (12.25) \wedge (12.26) \wedge (12.27) \wedge (12.28) \wedge (12.29)$. *Then we have*

$$\begin{gathered}
CIRC[\Sigma; Initiates, Terminates, Releases] \wedge CIRC[\Delta; Happens] \wedge \\
CIRC[\Theta; Ab_1] \wedge \Omega \wedge \Gamma \wedge \mathrm{DEC} \\
\vdash (\neg HoldsAt(DeterminingFluent(Device1), 0) \Rightarrow \\
HoldsAt(On(Device1), 1)) \wedge \\
(HoldsAt(DeterminingFluent(Device1), 0) \Rightarrow \\
\neg HoldsAt(On(Device1), 1))
\end{gathered}$$

Proof. See Exercise 12.1. ■

12.5 DEFAULT EVENTS AND PROPERTIES

Default reasoning can be used to represent events that occur by default, as well as properties that are true or false by default.

12.5.1 DEFAULT EVENTS

A default event is represented by an axiom of the form

$$\gamma \wedge \neg\rho(\tau_1, \ldots, \tau_n, \tau) \Rightarrow Happens(\alpha, \tau)$$

where γ is a condition, ρ is an abnormality predicate symbol of arity $n+1$, τ_1, ..., τ_n are terms, τ is a timepoint term, and α is an event term.

We may specify that normally a watch beeps every 3600 timepoints, starting at timepoint 0:

$$\begin{gathered}
Remainder(t, 3600) = 0 \wedge \neg Ab_1(w, t) \Rightarrow \\
Happens(Beep(w), t)
\end{gathered}$$

Using the alarm clock axiomatization in Section 4.1.1, we may rewrite the trigger axiom (4.8) as follows:

$$\begin{gathered} HoldsAt(AlarmTime(c,t),t) \wedge \\ HoldsAt(AlarmOn(c),t) \wedge \\ \neg Ab_1(c,t) \Rightarrow \\ Happens(StartBeeping(c),t) \end{gathered}$$

This states that normally a clock starts beeping when its alarm time is the current time.

12.5.2 DEFAULT PROPERTIES

A default property such as a typical color or location is represented by a formula of the form

$$\neg\rho(\tau_1, \ldots, \tau_n, \tau) \Rightarrow HoldsAt(\beta, \tau)$$

where ρ is an abnormality predicate symbol of arity $n+1$, $\tau_1, \ldots, \tau_n$ are terms, τ is a timepoint term, and β is a fluent term.

Suppose we have an apple sort with variable a. We represent the typical color of apples as follows:

$$\neg Ab_1(a,t) \Rightarrow HoldsAt(Red(a),t)$$

We add cancellation axioms representing when apples are not red:

$$\begin{gathered} HoldsAt(GrannySmith(a),t) \Rightarrow Ab_1(a,t) \\ HoldsAt(Rotten(a),t) \Rightarrow Ab_1(a,t) \\ HoldsAt(NewtownPippin(a),t) \Rightarrow Ab_1(a,t) \end{gathered}$$

Suppose we have a living room sort with variable l and a television sort with variable v. We represent the typical location of a television as follows:

$$\neg Ab_1(l,t) \Rightarrow \exists v\, HoldsAt(InRoom(v,l),t)$$

BIBLIOGRAPHIC NOTES

Frames and inheritance networks

Minsky (1974) proposed a knowledge structure called *frames* for representing typical situations. The slots of a frame have default assignments that "can be easily displaced by new items that fit better the current situation" (p. 2). A number of framelike knowledge representation languages with defaults have been developed (Bobrow & Winograd, 1977; Charniak, Riesbeck, & McDermott, 1980, chap. 15; Roberts & Goldstein, 1977). Frames are reviewed by Brachman and Levesque (2004, chap. 8). Work on semantic networks (Ceccato, 1966; Quillian, 1968; Winston, 1970) and frames led to the development of inheritance networks (Fahlman, 1979; Horty, Thomason, & Touretzky, 1990; Stein, 1992), which can be used for default reasoning; inheritance networks are reviewed by Brachman and Levesque (2004, chap. 10).

Benchmark problems

Three benchmark problems were important in the development of logic-based methods for default reasoning: the Tweety example, the Nixon diamond, and the Yale shooting scenario. Reiter (1978, p. 215; 1980b, pp. 82-83) introduced the Tweety example, which can be traced back to work on semantic networks (A. M. Collins & Quillian, 1969). The example is as follows. All birds fly except for penguins, which do not fly. If we are told that Tweety is a bird, then we should conclude that Tweety flies. But if we are told that Tweety is a penguin, then we should retract the conclusion that Tweety flies and conclude that Tweety does not fly. Reiter (1980b) introduced default logic to deal with examples of default reasoning such as this. Lifschitz (1999) discusses the success of default logic.

The Nixon diamond, introduced by Reiter and Criscuolo (1981, p. 274), is an example of conflicting defaults. The example is as follows. Quakers are typically pacifists, and Republicans are typically not pacifists. If we are told that John is both a Quaker and a Republican, what shall we conclude? The Yale shooting scenario is discussed in Section 5.2.1.

Logic-based methods for default reasoning are reviewed by Brachman and Levesque (2004) and Russell and Norvig (2009). See also the discussion of non-monotonic reasoning methods in the Bibliographic notes of Chapter 2.

Methods based on circumscription

McCarthy (1977) proposed circumscription as a “candidate for solving the qualification problem” (p. 1040). McCarthy (1980, pp. 36-37) introduced the method of circumscription of abnormality for default reasoning using a predicate called *prevents*. McCarthy (1984a, 1986, 1987) and Grosof (1984) further developed the method and introduced the *ab* predicate. Levesque (1981, chap. 6) proposed a similar method involving *prototypical predicates*, or the negations of abnormality predicates. The prototypical predicate $\nabla \rho : \mathrm{i}(x)$ represents that x is prototypical with respect to the ith default property of ρ. For example, we can write the following:

$$\nabla Apple : 1(x) \Rightarrow Red(x)$$
$$GrannySmith(x) \Rightarrow \neg \nabla Apple : 1(x)$$

The analogous formulas using abnormality predicates are:

$$Apple(x) \wedge \neg Ab_1(x) \Rightarrow Red(x)$$
$$GrannySmith(x) \Rightarrow Ab_1(x)$$

Defaults are applied by a function $\Delta Ext(m)$, which extends a knowledge base m with formulas such as

$$Apple(x) \wedge \neg K \neg \nabla Apple : 1(x) \Rightarrow \nabla Apple : 1(x)$$

which states that if x is an apple and it is not known that x is not prototypical with respect to the first default property of apples, then x is prototypical with respect to this property.

The method of default reasoning used in this chapter is one of a number of proposed methods based on circumscription, which are summarized in Table 12.1. For each method, the table lists the techniques used in the method, an example of a

Table 12.1 Default Reasoning Methods Based on Circumscription[a]

Method	Example	CIRC	Result
(A) var (Lifschitz, 1994; McCarthy, 1986)	$\Phi =$ $(P(x) \wedge \neg Ab(x) \Rightarrow Q(x)) \wedge$ $\neg Q(A) \wedge P(A) \wedge P(B) \wedge$ $A \neq B$	$CIRC[\Phi; Ab; Q] \vdash$ $Ab(x) \Leftrightarrow x = A$	$Q(B)$
(B) cancellation + var (Lifschitz, 1994; McCarthy, 1986)	$\Phi =$ $(P(x) \wedge \neg Ab(x) \Rightarrow Q(x)) \wedge$ $Ab(A) \wedge P(A) \wedge P(B) \wedge$ $A \neq B$	$CIRC[\Phi; Ab; Q] \vdash$ $Ab(x) \Leftrightarrow x = A$	$Q(B)$
(C) conjunction + var (Lifschitz, 1994; McCarthy, 1986)	$\Phi =$ $(P(x) \wedge \neg Ab_1(x) \Rightarrow Q(x)) \wedge$ $(R(x) \wedge \neg Ab_2(x) \Rightarrow \neg Q(x)) \wedge$ $P(A) \wedge R(A)$	$CIRC[\Phi; Ab_1; Q] \wedge$ $CIRC[\Phi; Ab_2; Q] \vdash$ $(Ab_1(x) \Leftrightarrow P(x) \wedge$ $R(x) \wedge \neg Ab_2(x)) \wedge$ $(Ab_2(x) \Leftrightarrow P(x) \wedge$ $R(x) \wedge \neg Ab_1(x))$	
(D) cancellation + conjunction + var (Lifschitz, 1994; McCarthy, 1986)	$\Phi =$ $(P(x) \wedge \neg Ab_1(x) \Rightarrow Q(x)) \wedge$ $(R(x) \wedge \neg Ab_2(x) \Rightarrow \neg Q(x)) \wedge$ $(R(x) \Rightarrow P(x)) \wedge R(A) \wedge$ $P(B) \wedge \neg R(B) \wedge A \neq B \wedge$ $(R(x) \Rightarrow Ab_1(x))$	$CIRC[\Phi; Ab_1; Q] \wedge$ $CIRC[\Phi; Ab_2; Q] \vdash$ $(Ab_1(x) \Leftrightarrow R(x)) \wedge$ $\neg Ab_2(x)$	$Q(B)$ $\neg Q(A)$
(E) conjunction + priorities + var (Lifschitz, 1994; McCarthy, 1986)	$\Phi =$ $(P(x) \wedge \neg Ab_1(x) \Rightarrow Q(x)) \wedge$ $(R(x) \wedge \neg Ab_2(x) \Rightarrow \neg Q(x)) \wedge$ $(R(x) \Rightarrow P(x)) \wedge R(A) \wedge$ $P(B) \wedge \neg R(B) \wedge A \neq B$	$CIRC[\Phi; Ab_1; Q] \wedge$ $CIRC[\Phi; Ab_2; Q, Ab_1]$ $\vdash (Ab_1(x) \Leftrightarrow R(x)) \wedge$ $\neg Ab_2(x)$	$Q(B)$ $\neg Q(A)$
(F) cancellation + conjunction + filtering + parallel = forced separation (Shanahan, 1997b)	$\Phi_1 = (P(x) \Rightarrow Ab_1(x)) \wedge$ $(Q(x) \Rightarrow Ab_2(x))$ $\Phi_2 = (R(x) \Rightarrow Ab_3(x))$ $\Phi_3 =$ $(S(x) \wedge \neg Ab_1(x) \Rightarrow T(x)) \wedge$ $(U(x) \wedge \neg Ab_2(x) \Rightarrow V(x)) \wedge$ $(W(x) \wedge \neg Ab_3(x) \Rightarrow Y(x)) \wedge$ $S(A) \wedge \neg P(A)$	$CIRC[\Phi_1; Ab_1, Ab_2] \wedge$ $CIRC[\Phi_2; Ab_3] \wedge$ $\Phi_3 \vdash$ $(Ab_1(x) \Leftrightarrow P(x)) \wedge$ $(Ab_2(x) \Leftrightarrow Q(x)) \wedge$ $(Ab_3(x) \Leftrightarrow R(x))$	$T(A)$

[a] *cancellation = cancellation of inheritance axioms; conjunction = conjunction of circumscriptions; parallel = parallel circumscription; var = circumscription with varied constants.*

domain theory, the circumscriptions of the domain theory, and the results entailed by the circumscriptions. Methods (A) and (B) address default reasoning problems such as the Tweety example. Methods (C), (D), and (E) address problems of conflicting defaults such as the Nixon diamond. Method (F) was developed to deal with problems involving action and change such as the Yale shooting scenario.

Methods (A) through (E) use circumscription with varied constants (Lifschitz, 1994, sec. 2.3). In basic circumscription (defined in Section A.7), the extension of a predicate is minimized without allowing the extensions of other predicates to vary. The extensions of other predicates are fixed. In *circumscription with varied constants*, the extension of one predicate is minimized while allowing the extensions of certain other predicates to vary. If Γ is a formula containing the predicate symbols $\rho, \psi_1, \ldots, \psi_n$, then the circumscription of ρ in Γ with $\psi_1, \ldots, \psi_n$ varied, written $CIRC[\Gamma; \rho; \psi_1, \ldots, \psi_n]$, is the formula of second-order logic

$$\Gamma \wedge \neg\exists \phi, \nu_1, \ldots, \nu_n\, (\Gamma(\phi, \nu_1, \ldots, \nu_n) \wedge \phi < \rho)$$

where ϕ and $\nu_1, \ldots, \nu_n$ are distinct predicate variables with the same arities and argument sorts as ρ and $\psi_1, \ldots, \psi_n$, respectively, and $\Gamma(\phi, \nu_1, \ldots, \nu_n)$ is the formula obtained from Γ by replacing each occurrence of ρ with ϕ and each occurrence of ψ_i with ν_i for each $i \in \{1, \ldots, n\}$.

Methods (C) through (F) use conjunctions of circumscriptions. Method (C) adopts the skeptical stance that, because the defaults about $Q(A)$ conflict, neither $Q(A)$ nor $\neg Q(A)$ should be concluded.

Methods (B), (D), and (F) use cancellation of inheritance axioms (Genesereth & Nilsson, 1987, p. 129; Lifschitz, 1994, sec. 4.1; McCarthy, 1986, p. 93), which are called cancellation axioms in this chapter. Antoniou (1997, pp. 158-159) and Brewka, Dix, and Konolige (1997, pp. 18-19) criticize the use of cancellation of inheritance axioms on elaboration tolerance grounds. Consider the example given for (D). Notice that whenever we add an axiom $\rho_1(x) \wedge \neg\rho_2(x) \Rightarrow \neg Q(x)$ to the domain theory, where ρ_1 is a predicate symbol and ρ_2 is an abnormality predicate symbol, we must also remember to add a cancellation of inheritance axiom $\rho_1(x) \Rightarrow Ab_1(x)$. But also notice that this is not the only possible order of elaboration of the domain theory. After writing the axiom $P(x) \wedge \neg Ab_1(x) \Rightarrow Q(x)$, we might then think of a situation in which this default does not apply. We would then write a cancellation axiom $\rho_1(x) \Rightarrow Ab_1(x)$. Having canceled the default, we would then proceed to write an axiom describing what the new default is in this situation: $\rho_1(x) \wedge \neg\rho_2(x) \Rightarrow \neg Q(x)$.

Antoniou (1997) and Brewka, Dix, and Konolige (1997) advocate replacing cancellation of inheritance axioms with prioritized circumscription. Method (E) replaces the cancellation of inheritance axiom in method (D) with the use of priorities on abnormality predicate symbols. The abnormality predicate symbol Ab_2 is given a higher priority than Ab_1 by allowing Ab_1 to vary in the circumscription of Ab_2. This method can be generalized. For example, we may specify that Ab_3 has a higher priority than Ab_2, which has a higher priority than Ab_1, as follows:

Table 12.2 Default Reasoning Using Basic Circumscription and Filtering

Method	Example	CIRC	Result
(B′) cancellation + filtering	$\Phi_1 = Ab(A)$ $\Phi_2 =$ $(P(x) \land \neg Ab(x) \Rightarrow Q(x)) \land$ $P(A) \land P(B) \land A \neq B$	$CIRC[\Phi_1; Ab] \land$ $\Phi_2 \vdash$ $Ab(x) \Leftrightarrow x = A$	$Q(B)$
(D′) cancellation + conjunction + filtering	$\Phi_1 = (R(x) \Rightarrow Ab_1(x))$ $\Phi_2 = (\bot \Rightarrow Ab_2(x))$ $\Phi_3 =$ $(P(x) \land \neg Ab_1(x) \Rightarrow Q(x)) \land$ $(R(x) \land \neg Ab_2(x) \Rightarrow \neg Q(x)) \land$ $(R(x) \Rightarrow P(x)) \land R(A) \land$ $P(B) \land \neg R(B) \land A \neq B$	$CIRC[\Phi_1; Ab_1] \land$ $CIRC[\Phi_2; Ab_2] \land$ $\Phi_3 \vdash$ $(Ab_1(x) \Leftrightarrow R(x)) \land$ $\neg Ab_2(x)$	$Q(B)$ $\neg Q(A)$

$$
\begin{gathered}
CIRC[\Phi; Ab_1; Q] \land \\
CIRC[\Phi; Ab_2; Q, Ab_1] \land \\
CIRC[\Phi; Ab_3; Q, Ab_1, Ab_2]
\end{gathered}
$$

Etherington (1988, p. 47) criticizes the use of priorities on the grounds that it is difficult to determine what the priorities should be.

Method (F) incorporates the technique of filtering (Doherty, 1994; Doherty & Łukaszewicz, 1994; Sandewall, 1989b, 1994), in which the domain theory is partitioned and only some parts of the partition are circumscribed. Filtering is related to the process of delimited completion (Genesereth & Nilsson, 1987, p. 132). Method (F), which is used in this chapter, is called forced separation in Shanahan (1996, 1997b). In forced separation, a domain theory is partitioned into formulas $\Phi_1, \ldots, \Phi_n$, disjoint sets of predicates are circumscribed in Φ_i for each $i \in \{1, \ldots, n-1\}$, and Φ_n is not circumscribed. The combination of cancellation, conjunction, and filtering from method (F) can be used to solve problems addressed by methods (B) and (D), as shown in Table 12.2.

Our treatment of the qualification problem is patterned after that of E. Giunchiglia, Lee, Lifschitz, McCain, and Turner (2004, pp. 67-68). Doherty and Kvarnström (1997) propose the distinction between strong and weak qualifications. See also the discussion of Thielscher (2001, pp. 27-29). Williams (1997) describes a framework for modeling revision of beliefs.

EXERCISES

12.1 Prove Proposition 12.3.

12.2 Formalize the slippery block scenario described by Thielscher (2001, p. 27). A surface is a block or a table. Normally when a block is moved from one surface to another, the block ends up on the second surface. If the block is slippery, however, then the block may (or may not) end up on the table rather than the second surface.

12.3 Formalize Lifschitz's (2003) problem of choosing a place for dinner. Each night, Peter eats at either the Chinese restaurant or the Italian restaurant. On Mondays, the Italian restaurant is closed. Peter prefers not to eat at the same restaurant two nights in a row. Use the formalization to determine where Peter eats on Sundays.

12.4 Formalize the Russian airplane hijack scenario elaborated by Kvarnström and Doherty (2000a) based on a suggestion from Lifschitz (1997). Boris, Dimiter, and Erik are at their respective homes getting ready for a business trip. Dimiter gets drunk. Erik grabs a comb. They each stop off at the office on their way to the airport. While at the office, Boris picks up a gun. They each try to board a plane, which then flies to Stockholm. Normally, if a person is at the airport, then the person can board the plane. However, people carrying guns are not permitted to board the plane, and drunk people may or may not be permitted to board the plane. The problem is then to show that Erik and the comb will be in Stockholm, Boris will still be at the airport, and Dimiter may or may not be at Stockholm.

12.5 (Research problem) In Section 11.2, we introduce the predicates $Like(a, o)$ and $Dislike(a, o)$. Formalize the default rules given by Heider (1958, chap. 7) regarding the ways people tend to evaluate or feel about people or things.