Default Reasoning Through Belief Revision Strategy

Chern H. Seet

Information Technology Institute
National Computer Board
71 Science Park Drive
Singapore 0511

Abstract

The thesis of this paper is that default reasoning can be accomplished rather naturally if an appropriate strategy of belief revision is employed. The idea is based on the premise that new beliefs introduced into a situation change the structure of current beliefs to accomodate the new beliefs as exceptions. It is easy to characterise these exceptions in beliefs if we extend the belief language to include some modal operator and prefix the exceptions with the operator. This serves to make the exceptions syntactically explicit, which can then be processed in a routine way by a default reasoning theorem prover.

I. Introduction

Default reasoning tries to model the phenomenon of human reasoning that makes us jump to conclusions that are typical of what we know. Paraphrasing a classical example, a case of default reasoning is this: If we know that *Tweety* is a bird, and that birds in general can fly, then we are led to conclude that *Tweety* can fly if there is no evidence of an exception such as the fact that *Tweety* might be a penguin. There are many approaches that have been taken to characterise default reasoning. Among them are the following three: The *nonmonotonic logic* of [McDermott and Doyle, 1980], the *default logic* of [Reiter, 1980], and the *circumscription* approach of [McCarthy, 1980].

Belief revision basically concerns the maintenance of a knowledge base to reflect changes made to it. Some major efforts in this field are the truth (or reason) maintenance systems of [Doyle, 1979] and [de Kleer, 1986], where the emphasis is on justifying beliefs. In this paper our emphasis is on devising a strategy of modifying beliefs (given as a set of sentences) syntactically in a manner that will support default reasoning.

The thesis of this paper is that default reasoning can be accomplished rather naturally if such a strategy of belief revision is employed. The idea is based on the premise that new beliefs introduced into a situation (or *world*) changes the structure of current beliefs to accomodate the new beliefs as *exceptions*. It is easy to characterise these exceptions in beliefs if we extend the belief language to include some modal operator and prefix the exceptions with the operator. This serves to make the exceptions of a belief syntactically explicit, which can then be processed in a routine way by a default reasoning theorem prover. A fundamental difference in our approach to default reasoning from those of McDermott and Doyle, Reiter, and McCarthy is the syntactic explicitness we confer on exceptions. In their approaches, exceptions are not directly distinguisable from defaults in the set of sentences denoting beliefs.

A broad description of our approach is this: First we establish the concept of *worlds* of beliefs, where a world can be regarded as a state of beliefs or knowledge of an intelligent reasoning agent (The agent would "move" from world to world upon the acquisition of new beliefs). Associated with a world are certain belief-related entities to be described shortly. Second, we use a modal operator in our language of beliefs to represent the exceptions to beliefs. This is described in Section II, where we also outline a method to deduce "default theorems" from a set of beliefs. With this language, we can then manipulate the beliefs to reflect new exceptions upon acquiring new beliefs. This is outlined in Section III.

We take beliefs to be characterised by a type of clausal formulas which we shall introduce in the next section, and regard a clause as a unit of belief. With each world we associate these three entities:

β — the *new belief*
B_C — the *current beliefs*
B_F — the *future beliefs*

B_C represents the set of current beliefs — those things that the agent believed when he first moved into the current world. β represents the new belief acquired by the agent in the current world after an unspecified period of stay. For simplicity, we assume new beliefs come in one unit at a time (the unit is a clause). Also, we shall not be concerned with how the new belief is acquired since this depends on what the agent is. Two scenarios can prevail when β is acquired by the agent:

1. The new belief is consistent with the current beliefs. We regard β as an *additional information* over the agent's current beliefs, and add it to B_C to yield his future beliefs B_F.
2. The new belief is inconsistent with the current beliefs. We regard β as a correction of what the agent believed, and revise B_C to B_F in a manner to be elaborated later.

The acquisition of the new belief β heralds the agent's transition into a new world whose current belief is the B_F just derived from the previous world. The cycle of hopping from world to world continues indefinitely in this fashion.

The first scenario above is straightforward, where belief revision is more akin to *belief refinement* since B_F is more specific than B_C in its characterisation of what the agent believes. The second scenario is where all the challenges lie, and so we shall concentrate on it for the rest of the paper. We now turn our attention to the belief language and how default reasoning is employed on beliefs.

II. Default Reasoning on Beliefs

We call an expression of the form $\Box\varphi$ where φ is some first-order formula a *$\Box$-term*. A *clause* is a disjunction of literals and $\Box$-terms. Free variables in clauses are assumed to be universally quantified.

Example: $\neg Bird(x) \vee Fly(x) \vee \Box Penguin(x)$ is a clause. $\Box$-terms are used to denote exceptions, in the sense that $\Box Penguin(x)$ is the exception to the expression that all birds can fly. For a more natural reading, we shall often express clauses in non-clausal form such as $Bird(x) \wedge \neg\Box Penguin(x) \rightarrow Fly(x)$.

A *new belief* is a clause without $\Box$-terms, i.e., we assume new beliefs to be exception-free. A *belief* is a clause, so B_C and B_F are sets of clauses which may contain $\Box$-terms.

As we shall see, the proof theory for beliefs is simple and conceptually appealing. There are two notions of logical consequence:

$B \vdash \varphi$ which informally means "Strictly speaking, φ is true"

$B \circ\!\vdash \varphi$ which informally means "By default, φ is true"

where B is a set of beliefs and φ is a first-order formula.

1. Definition of $B \vdash \varphi$

For a set B of beliefs, define $strict(B)$ to be the set of clauses transformed from B by replacing each $\Box$-term $\Box\psi$ by ψ (i.e., $strict(B)$ is a theory where exceptions are not distinguishable). We then define $B \vdash \varphi$ iff φ is a theorem of $strict(B)$ according to some standard system of first-order provability (such as resolution refutation). We shall also use the turnstile $\vdash$ for standard first-order provability since there should be no danger of confusion. Hence $B \vdash \varphi$ iff $strict(B) \vdash \varphi$.

2. Definition of $B \circ\!\vdash \varphi$

A term like $\Box\psi$ in a set of beliefs B is interpreted to mean that ψ is provable from B, i.e., $\Box\psi$ iff $B \vdash \psi$. Informally, we define $B \circ\!\vdash \varphi$ iff φ is provable in a deduction system incorporating the rules of first-order logic and the meta-inference rule:

From $B \vdash \psi$, *infer* $\Box\psi$

This can be defined more rigorously using a system of modal logic, but we shall not do this since the broad concepts behind the meaning of $\circ\!\vdash$ should be clear. Knowing this meaning, we can adapt the resolution refutation method to incorporate the meta-inference rule using the following strategy:

1. $\Box$-terms are not unifiable with any literal.

   Example: $\Box Penguin(x)$ and $\neg Penguin(Penny)$ are not unifiable.

2. The free variables of $\Box$-terms in a clause are affected by substitutions when that clause is used during resolution.

   Example: Resolving the unit clause $\neg Pac$ with the clause $Pax \vee \Box(\exists y(Qxy \wedge \neg Pyb)) \vee Rxb$ yields the resolvent $\Box(\exists y(Qay \wedge \neg Pyb)) \vee Rab$ where the free variable x has been substituted with a.

3. A $\Box$-term $\Box\varphi$ that has no more free variables is removed from the clause if, by "spawning" another resolution refutation process to prove φ from B, the prove fails. If $\Box\varphi$ was the last item in the clause and the proof failed, then as usual the resolvent is the empty clause, $\Box$ say, which denotes the success of the refutation.

The rationale for this strategy is that we try to *refute* terms like $\Box\varphi$, i.e., disprove $B \vdash \varphi$, in the same way that we try to refute ordinary literals through unification. Hence the $\Box$-terms may be viewed as "literals" where the "unification" process is the spawning of a separate resolution refutation process.

Example: The resolvent $\Box\exists y(Qay \wedge \neg Pyb) \vee Rab$ obtained above can be reduced to Rab if we can refute $B \vdash \exists y(Qay \wedge \neg Pyb)$.

Example: Let B be these beliefs:

$$Bird(x) \wedge \neg\Box Penguin(x) \rightarrow Fly(x)$$
$$Penguin(x) \rightarrow \neg Fly(x)$$
$$Penguin(x) \rightarrow Bird(x)$$
$$Bird(Tweety)$$
$$Penguin(Penny)$$

Then $strict(B)$ is the same theory but with the first clause replaced by $Bird(x) \wedge \neg Penguin(x) \rightarrow Fly(x)$. We then have the following (free variables assumed universally quantified):

(a) $B \vdash Bird(x) \wedge \neg Penguin(x) \rightarrow Fly(x)$
Birds that are not penguins can fly

(b) $B \nvdash Bird(x) \rightarrow Fly(x)$
We cannot conclude that all birds can fly, or, *strictly speaking*, not all birds can fly

(c) $B \circ\!\vdash Bird(x) \rightarrow Fly(x)$
By default, all birds can fly, or, *generally speaking*, all birds can fly

(d) $B \nvdash Fly(Tweety)$
Strictly speaking, we cannot conclude that *Tweety* can fly

(e) $B \circ\!\vdash Fly(Tweety)$
By default, (or *probably*) *Tweety* can fly

(f) $B \circ\!\not\vdash Fly(Penny)$
We cannot conclude that by default *Penny* can fly

(h) $B \circ\!\not\vdash Fly(Chirpy)$
We cannot conclude that by default *Chirpy* can fly

We now explain the derivation of those cases above involving $\circ\!\vdash$.

Case (c). To prove $B \circ\!\vdash Bird(x) \rightarrow Fly(x)$ by resolution refutation, we resolve $Bird(k)$ and $\neg Fly(k)$ (k is a skolem constant) with the first clause of B to get the resolvent $\Box Penguin(k)$. Now we spawn another refutation process to refute $B \vdash Penguin(k)$, which succeeds (note that the skolem constant k does not unify with *Penny*). So the original resolvent $\Box Penguin(k)$ is reduced to the empty clause $\Box$, and we have thus proved $B \circ\!\vdash Bird(x) \rightarrow Fly(x)$.

Case (e). To prove $B \circ\!\vdash Fly(Tweety)$, we resolve $\neg Fly(Tweety)$ with the first clause of B to get $\neg Bird(Tweety) \vee \Box Penguin(Tweety)$. Spawning a refutation process to refute $B \vdash Penguin(Tweety)$ succeeds, so our resolvent is reduced to $\neg Bird(Tweety)$. This finally resolves with $Bird(Tweety)$ to yield the empty clause.

Case (f). Similarly to case (e), we encounter the resolvent $\neg Bird(Penny) \vee \Box Penguin(Penny)$.
Further resolution with other clauses yield the resolvent $\Box Penguin(Penny)$. But now we cannot reduce this further since we cannot refute $B \vdash Penguin(Penny)$. Hence $B \circ\!\not\vdash Fly(Penny)$.

Case (g). Similarly to the above two cases, we encounter the resolvent $\neg Bird(Chirpy) \vee \Box Penguin(Chirpy)$. This reduces to $\neg Bird(Chirpy)$ since we can refute $B \vdash Penguin(Chirpy)$. But we cannot refute $\neg Bird(Chirpy)$. Hence $B \circ\!\not\vdash Fly(Chirpy)$.

III. Belief Revision Strategy

In section I, we mentioned that belief revision occurs when the new belief β is inconsistent with the current beliefs B_C. In the context of $\circ\!\vdash$, we can take this to mean that belief revision occurs if $B \cup \{\beta\} \circ\!\vdash \varphi$ and $B \cup \{\beta\} \circ\!\vdash \neg\varphi$ for some formula φ. We shall assume this to be our policy for belief revision for the time being. Later we shall remark on a more relaxed policy. Before we proceed, we need some definitions.

The *unification condition* of a literal and its conjugate $Rt_1 \ldots r_n$ and $\neg Rt'_1 \ldots t'_n$ is the formula $(t_1 = t'_1 \wedge \ldots \wedge t_n = t'_n)$. We say two literals are *unifiable* if they are conjugates with a unification condition that does not include the equality $c = c'$ where c and c' are distinct constant symbols (i.e., we take $c \neq c'$ as an axiom for distinct c, c').

Example: The unification condition of $\neg Qzb$ and Qax is $(z = a \wedge b = x)$.

We assume that with each new belief β there is one specially designated literal denoted by u_β. This literal will be printed in bold, such as in $Bird(x) \rightarrow \mathbf{Fly(x)}$. This designation of u_β is to reflect the preferred way to express a new belief, where u_β is the consequent implied by the antecedent in an implication. This scheme is a natural one since there is normally a preferred way to express beliefs. For instance, the expression $Bird(x) \rightarrow \mathbf{Fly(x)}$ is more natural than the expression $\neg Fly(x) \rightarrow \neg\mathbf{Bird(x)}$.

If σ is a unification condition, then $\beta[\sigma]$ is the formula derived as follows: First transform the clause β to the clause β' by replacing u_β by $\neg\sigma$, then prefix β' by the universal quantifiers $\forall v_1 \ldots \forall v_n$ for each variable v_i in β' to obtain $\beta[\sigma]$.

Example: For the new belief $\beta = \neg Pxy \vee \mathbf{Qax}$ and the unification condition $\sigma = (z = a \wedge b = x)$,
$\beta[\sigma] = \forall x \forall y(\neg Pxy \vee \neg(z = a \wedge b = x))$.
Simplifying, we get $\beta[\sigma] \equiv \forall x \forall y \neg(Pxy \wedge z = a \wedge b = x)$
$\equiv \forall y \neg(Pby \wedge z = a) \equiv \neg\exists y(Pby \wedge z = a)$

Now we can explain the belief revision strategy. The clauses in the future beliefs B_F are:

- the new belief β, and
- clauses derived from B_C as follow:

  1 For each clause in B_C that does not contain literals unifiable with u_β, put the clause in B_F.

  2 For each remaining clause c in B_C, first transform it to c' in which each literal l of c that is unifiable with u_β has been replaced by $l \vee \Box\neg\beta[\sigma]$ (c' is therefore weaker than c). Put the resultant c' in B_F.

  (In both 1 and 2, "literals" mean ordinary literals, not those that occur in $\Box$-terms.)

Example: To illustrate case 2, assume $\neg Bird(x) \vee Fly(x)$ is is a clause in B_C. This clause would be transformed to the following before being put in B_F:

$\neg Bird(x) \vee Fly(x) \vee \Box Ostrich(x)$
if $\beta = \neg Ostrich(y) \vee \neg\mathbf{Fly(y)}$

$\neg Bird(x) \vee Fly(x) \vee \Box(x = Tweety)$
if $\beta = \neg\mathbf{Fly(Tweety)}$

$\neg Bird(x) \vee Fly(x) \vee \Box(Caged(Tweety) \wedge x = Tweety)$
if $\beta = \neg Caged(Tweety) \vee \neg\mathbf{Fly(Tweety)}$

$\neg Bird(x) \vee Fly(x) \vee \Box(\exists z\ HasAilment(x, z))$
if $\beta = HasAilment(y, z) \vee \neg\mathbf{Fly(y)}$

It can be verified that the B_F thus derived is consistent with β. To see this quickly, suppose B_C contains just $Fly(Tweety)$, and $\beta = \neg\mathbf{Fly(Tweety)}$. Then B_C and β are inconsistent. However, the transformed clause in B_F is $Fly(Tweety) \vee \Box(Tweety = Tweety)$, which is a valid formula. So B_F contains this valid formula and β, and is therefore trivially consistent with β (valid formulas can optionally be dropped from B_F since they provide no additional information). Thus, the transformation process serves to weaken the clauses of B_C to ensure consistency with β.

Example: If $\beta = \neg Ostrich(y) \vee \mathbf{Bird(y)}$ and B_C is

$\neg Bird(x) \vee Fly(x) \vee \Box Penguin(x)$
$\neg Ostrich(x) \vee \neg Fly(x)$
$Ostrich(Ossie)$

then B_C is inconsistent with β since

$B_C \cup \{\beta\} \circ\vdash Fly(Ossie)$, and
$B_C \cup \{\beta\} \circ\vdash \neg Fly(Ossie)$.

Using the belief revision strategy, B_F is

$\neg Bird(x) \vee Fly(x) \vee \Box Penguin(x) \vee \Box Ostrich(x)$
$\neg Ostrich(x) \vee \neg Fly(x)$
$Ostrich(Ossie)$
$\neg Ostrich(x) \vee Bird(x)$

Note that $B_F \circ\vdash \neg Fly(Ossie)$ but $B_F \circ\not\vdash Fly(Ossie)$. The latter can be demonstrated using the adapted resolution refutation method described earlier, where it will be found that the term $\Box Ostrich(Ossie)$ cannot be removed from the resolvent encountered.

At the beginning of this section, we remarked that belief revision occurs when B_C is inconsistent with β (all reference to consistency here is taken with respect to $\circ\vdash$). It seems natural to relax this condition to allow belief revisions to also occur under certain situations. For instance, in the B_C of the last example, if $Ostrich(Ossie)$ is not a clause in B_C, then although B_C would not be inconsistent with β, we would still like the belief revision to take place. The intuitive motivation behind this is the observation that for this reduced B_C, we have $B_C \cup \{\beta\} \cup \{\exists x Ostrich(x)\}$ as inconsistent. Generalizing this, a more relaxed policy for proceeding with belief revision is when $B_C \cup \{\beta\} \cup S$ is inconsistent, where S is a set of formulas of the form $\exists x_1 \ldots \exists x_n P(x_1, \ldots, x_n)$, where P is an n-ary predicate in B_C (i.e., every predicate in B_C has a "witness").

There are two points about the belief revision strategy that should not be missed. First, *any* clause in B_C can play the role of the default expression for the next world. Thus if β in the above example was $\neg Superbreed(x) \vee \mathbf{Fly(x)}$ (all superbreed birds can fly), then the future beliefs B_F would be

$\neg Bird(x) \vee Fly(x) \vee \Box Penguin(x)$
$\neg Ostrich(x) \vee \neg Fly(x) \vee \Box Superbreed(x)$
$Ostrich(Ossie)$
$\neg Superbreed(x) \vee Fly(x)$

This will allow us in the future world to deduce that ostriches generally cannot fly:

$B_F \circ \vdash Ostrich(x) \rightarrow \neg Fly(x)$,

unless they are superbreed birds, which can fly:

$B_F \vdash Superbreed(x) \rightarrow Fly(x)$.

The second point is that the strategy has a natural "undo" property. Suppose the above four clauses are now the current beliefs B_C, and the new belief is that superbreed birds are in fact abnormal, and so cannot fly, i.e., $\beta = \neg Superbreed(x) \vee \neg \mathbf{Fly(x)}$. First, observe that $B_C \not\vdash Ostrich(x) \rightarrow \neg Fly(x)$, i.e., we are unable to derive from B_C our past belief that, strictly speaking, ostriches cannot fly. Now, B_F is

$\neg Bird(x) \vee Fly(x) \vee \Box Penguin(x) \vee \Box Superbreed(x)$
$\neg Ostrich(x) \vee \neg Fly(x) \vee \Box Superbreed(x)$
$Ostrich(Ossie)$
$\neg Superbreed(x) \vee Fly(x) \vee \Box Superbreed(x)$
$\neg Superbreed(x) \vee \neg Fly(x)$

where the last clause is β, and the first and fourth clauses are weakened from the corresponding clauses in B_C. It can be shown that $B_F \vdash Ostrich(x) \rightarrow \neg Fly(x)$, i.e., we have recovered our past belief that, strictly speaking, ostriches cannot fly.

IV. Conclusion

This paper has given an illustration of our thesis that default reasoning can be accomplished rather naturally if an appropriate strategy of belief revision is employed. An assumption behind our model of default reasoning is that the beliefs in the very first world are consistent and free of exceptions (i.e., contains no $\Box$-terms). The consistency of these first beliefs ensures that B_C of every accessible future world will also be consistent. As we move from the first world, exceptions will be gradually incorporated into the beliefs as $\Box$-terms. This strategy of tagging exceptions with a modal operator is reminiscent of, yet fundamentally different from, McDermott and Doyle's use of the modal operator M, McCarthy's approach of designating a predicate to stand for abnormal circumstances (circumscription), and Reiter's default logic. The fundamental difference is that in our approach, the exceptions to defaults are syntactically explicit.

It seems that a disadvantage of our approach is that formulas tend to get longer and longer as more exceptions are acquired. We have not investigated whether this is a serious disadvantage, and we suspect that the syntactic complexity arising from the longer formulas merely makes explicit the theorem-proving complexity latent in other approaches to default reasoning.

We have not in this paper probed deeper into the relationship between our approach to default reasoning and the more classical approaches. Further study of this relationships will clarify many relevant issues and perhaps also uncover some intriguing facts. Finally, we remark that our approach also embodies an elegant illustration of non-monotonic logic: The theorems provable in B_C may not always be provable in B_F. These are the theorems that were previously inferred but now have to be retracted because they are inconsistent with new beliefs.

Acknowledgements

My thanks to the two anonymous referees for their very useful critique of an earlier draft of this paper.

References

[de Kleer, 1986]
de Kleer, J., "Extending the ATMS", *Artificial Intelligence*, 28, 1980.

[Doyle, 1979]
Doyle, J., "A Truth Maintenance System", *Artificial Intelligence*, 12, 1979.

[McCarthy, 1980]
McCarthy, J., "Circumpscription – A Form of Non-Monotonic Reasoning", *Artificial Intelligence*, 13, 1980.

[McDermmot and Doyle, 1980]
McDermott, D., and Doyle, J., "Non-Monotonic Logic", *Artificial Intelligence*, 13, 1980.

[Reiter, 1980]
Reiter, R., "A Logic for Default Reasoning", *Artificial Intelligence*, 13, 1980.