Default Reasoning with Contexts

Daniel B. Hunter, Daniel F. Bostwick

BAE Systems, Advanced Information Technologies
6 New England Executive Park
Burlington, Massachusetts 01803
daniel.hunter@baesystems.com
daniel.bostwick@baesystems.com

Abstract

We describe a system that combines default reasoning with contexts. Contexts are arranged in a hierarchy where more specific contexts represent revisions of the state of belief in more general contexts. We describe our algorithm for default reasoning in a context hierarchy and provide a translation of our representation into a default logic theory whose inferences agree with our algorithm. We conclude with a discussion of different notions of context and give a justification of our default rules when contexts are understood as states of belief.

Introduction

People apply default reasoning all the time; we fill in missing information about a particular situation based on our experience and knowledge of what is usually true. Applying this type of reasoning is useful as it saves us from having to re-obtain information that remains largely static across most similar situations. Default reasoning in formal systems is likewise useful; it saves us from re-representing information that does not (usually) change.

People also believe different things at different times and in different situations; we operate in different states of belief according to changes in the information we have. The ability to revise our states of belief is an essential part of living in a dynamic world. Representing different states of belief in an automatic inference system is also useful; it allows us to do 'what-if' reasoning and can have a positive impact on inference efficiency. In this paper we view contexts as states of belief and investigate how default reasoning and contexts interact in reasoning.

We describe an algorithm for default reasoning in contexts that we have implemented in an automated inference system. We then provide a translation of our representation into a formal theory of default reasoning and show that our algorithm agrees with the theory. Finally we justify the theory when contexts are understood as states of belief.

Default Reasoning with Contexts in AKS

We have designed and implemented the AIT Knowledge Server (AKS), a computationally-efficient, constraint-based knowledge server that provides a semantics richer than conventional frame system attribute/value relations [Minsky, 1975]. In addition, the AKS supports contexts, which partition the knowledge base. Contexts enable reasoning within a subset of the knowledge base, and they allow different parts of the knowledge base to be inconsistent with each other, facilitating "what if" reasoning. In the AKS, contexts are arranged in a single (tree-structured) inheritance hierarchy such that anything that is true in a context is also true in its subcontexts.

The expressions in our representation language that are relevant to default reasoning are:

- *instance(C, I, K)*: class K is the most specific superclass of instance I in context C
- *parent_class(C, H, K)*: class K is a most specific superclass of class H in context C
- *subcontext(C2, C1)*: context C1 is the most specific supercontext of context C2
- *direct_assignment(C, I, S, V)*: V is assigned to be a value of slot S on instance I in context C
- *default_value(C, K, S, D)*: D is the default value of slot S on class K in context C

Default Reasoning in AKS

The AKS also supports a form of default reasoning in which a slot on a class may be assigned a default value. Default values are inherited by subclasses and instances of the class;

these default values can be overridden by subclasses and instances. In the event that an instance does not specify a value for a slot, then the inherited default value for the slot, if specified, becomes the value of the instance's slot. Slots are not required to have any value assigned to them.

The value for a slot on an instance can either be assigned directly or inherited from a default value in a superclass. If slot S on instance I (notated as I.S) has no directly assigned value, we search upwards in the class hierarchy for a most specific superclass of I that has a default value for S. With contexts, the reasoning becomes more complicated. Consider Figure 1 where we have two contexts:

1. a context called *birds* in which we are agnostic as to whether or not birds can fly, but we know that penguins, in general, cannot
2. a subcontext called *birds_can_fly* in which we have modified our belief to be that birds, in general, can fly

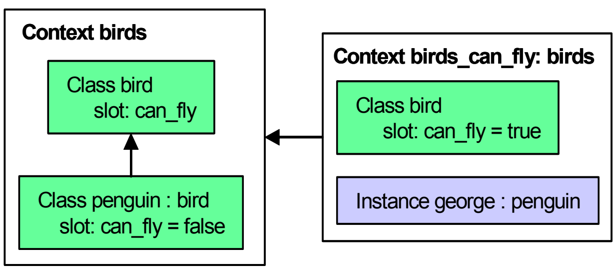

Figure 1. Class hierarchy over two contexts

The question is whether or not *george* can fly in the context *birds_can_fly*. Given a context C, an instance I of class K (where K is a most specific superclass of I), and a slot S on I, we use the following algorithm to determine the value of I.S in C. We use a special symbol *no_value_found* to indicate that no value has been assigned to I.S in C.

The function *value_of(C, I, S)* returns a value for slot S on instance I in context C. *value_of* first looks for a value that is directly assigned to I.S in C or any supercontexts of C. Failing that, *value_of* looks for a default value on slot S in the superclass of I in context C.

```
value_of(C, I, S ):
    V = direct(C, I, S)

    if V == no_value_found
        return default(C, K, S)
    else
        return V
```

direct(C, I, S) returns a value that is directly assigned to slot S on instance I in context C or any ancestor contexts of C.

```
direct( C, I, S ) :
    if direct_assignment(C, I, S,V ) then
        return V
    else
        if subcontext( C, Cp )
            return direct(Cp, I, S )
        else
            return no_value_found
```

The function *default(C, K, S)* returns a default value for slot S on class K in context C. The context hierarchy is searched before the class hierarchy. All ancestor contexts of C are searched for a default value on K.S before any superclasses of K are searched.

```
default(C, K, S ) :
    V = default_for_class(C, K, S )
    if V != no_value_found
        return V
    else
        for each Kp where parent_class(C, K, Kp )
            V = default(C, Kp, S )
            If V != no_value_found
                return V
        return no_value_found
```

The function *default_for_class(C, K, S)* returns a default value for slot S on class K in context C. *default_for_class* does not examine any superclasses of K; it searches only in context C and its supercontexts for a default value on K.S.

```
default_for_class(C, K, S ) :
    if default_value(C, K, S, V )
        return V
    else
        if subcontext( C, Cp )
            return default_for_class(Cp, K, S)
        else
            return no_value_found
```

For the situation illustrated in Figure 1 this algorithm gives the result:

```
value_of( birds_can_fly, george, can_fly ) = false
```

Justification of Default Reasoning Mechanism

[Etherington, 1988] provides a translation of a class inheritance hierarchy with exceptions into a set of default rules in Reiter's system of default logic. A similar translation can be given for our system of default reasoning.

In Reiter's system of default logic [Reiter, 1980], a *default theory* is a pair Δ = <D, W>, where W is a set of first-order formulas and D is a set of rules of the form:

$$\frac{\alpha(\bar{x}) : \beta(\bar{x})}{\chi(\bar{x})}$$

where $\alpha(\bar{x}), \chi(\bar{x})$, and $\beta(\bar{x})$ are first-order formulas whose free variables are among $\bar{x}$.

W is the set of facts and D provides a means of drawing tentative conclusions. The intuitive meaning of the above default rule is that if $\alpha(\bar{a})$ is known and it is consistent to believe $\beta(\bar{a})$, then $\chi(\bar{a})$ may be inferred. ($\bar{a}$ is a sequence of individual constants replacing the variables $\bar{x}$.)

A more formal interpretation of a default theory is provided by the notion of an *extension*. An extension of a default theory <D, W> is a set E of formulas that is a minimal fixed point[1] of an operator Γ on sets of formulas satisfying:

1. $W \subseteq \Gamma(S)$
2. $\Gamma(S)$ is closed under logical consequence
3. Given $\alpha(\bar{x}):\beta(\bar{x})/\chi(\bar{x}) \in D$, if $\alpha(\bar{a}) \in \Gamma(S)$ and $\neg\beta(\bar{a}) \notin S$, then $\chi(\bar{a}) \in \Gamma(S)$

An extension for a default theory is often regarded as a set of propositions constituting an "acceptable" set of beliefs given the theory.

A default rule with $\beta(\bar{x}) = \chi(\bar{x})$ is said to be *normal*; one with $\beta(\bar{x}) = \chi(\bar{x}) \wedge \varphi(\bar{x})$ for some $\varphi(\bar{x})$ is *semi-normal*. Our translation will always result in either normal or semi-normal default rules.

If Σ is a set of AKS assertions describing an inheritance hierarchy with contexts, we give a translation of Σ into a default theory <D(Σ), W(Σ)> as follows:

1. If instance(C, I, A) ∈ Σ, then C → A(I) ∈ W(Σ)
2. If parent_class(C, B, A) ∈ Σ, then (x)(C∧B(x) → A(x)) ∈ W(Σ)
3. If subcontext(C2, C1) ∈ Σ, then C2 → C1 ∈ W(Σ)
4. If direct_assignment(C,I,S,V) ∈ Σ, then the default $C{:}S(I,V) \wedge \neg C_1 \wedge \ldots \neg C_k / S(I,V) \in D(\Sigma)$, where the C_i are *nearest* subcontexts of C such that direct_assignment(C_i, I, S, V′) ∈ Σ for some V′ (i.e. there is no context between C and C_i in which a direct assignment to S for I is made).
5. If default_value(C, K, S, D) ∈ Σ, then $C \wedge K(x){:}S(x,D) \wedge \neg E_1 \wedge \ldots \neg E_k / S(x,D) \in D(\Sigma)$, where the E_i are all the *exceptions* (defined below) to default_value(C, K, S, D).

The exceptions to default_value(C, K, S, D) are the following:

i. If direct_assignment(C′, I, S, V) ∈ Σ, where C′ is C or a subcontext of C and I is an instance of K, then $(C' \wedge x{=}I)$ is an exception.

ii. If direct_assignment(C′, I, S, V) ∈ Σ, where C′ is a supercontext of C and I is an instance of K, then $x{=}I$ is an exception.

iii. If default_assignment(C′, H, S, D′) ∈ Σ, where C′ is a subcontext of C and H is K or a subclass of K, then C′∧H(x) is an exception.

iv. If default_assignment(C′, H, S, D′) ∈ Σ, where C′ is a supercontext of C and H is a subclass of K, then H(x) is an exception.

Given this translation, the algorithm for default reasoning described in the previous section can be shown to agree with the conclusions derivable from <D(Σ), W(Σ)>. More precisely, we have the following theorem:

Theorem 1. *If AKS infers conclusion P from default theory <D(Σ), W(Σ)>, then P is in an extension of <D(Σ), W(Σ)>*[2].

What Contexts Represent

Applying the translation to the example in the previous section yields the following default theory:

W = { birds_can_fly → birds, birds_can_fly → penguin(george) }

D = { birds∧penguin(X) : can_fly(X,false)/can_fly(X,false),

birds_can_fly∧bird(X) : can_fly(X,true) ∧¬penguin(X)/

can_fly(X,true) }

It is straightforward to show that can_fly(george,false) is derivable from birds_can_fly.

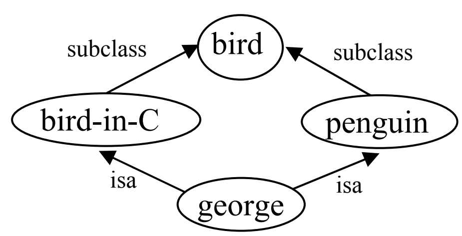

Figure 2. Avian inheritance hierarchy.

As the above default theory shows, we regard penguins as exceptions to birds flying even in subcontexts of the context in which penguins are declared not to fly by default; yet we don't regard birds in the context birds_can_fly as special kinds of birds whose default properties can override those of other subclasses of bird. This introduces an asymmetry in the treatment of subclasses and subcontexts. We might have regarded contexts as defining special subclasses of the classes defined within them, in which case we could

[1] X is a fixed point of an operator O if O(X) = X.

[2] For reasons of space, proofs could not be included in this paper. They are available upon request from the authors.

represent the entire class/context hierarchy in the previous section in terms of a single class hierarchy with no explicit contexts as shown in Figure 2. The context information is embedded in a special subclass of bird, the class bird-in-C (where C stands for a subcontext of the context birds).

If the default flying status for a bird-in-C is that it flies, then, we cannot arrive at a clear conclusion about George's flying ability. (This example is isomorphic to the notorious "Nixon diamond" [Touretzky, 1984].)

We wish to argue that whether or not any inference can be made about George's flying abilities given the above default rules depends upon how contexts are interpreted. [Akman and Surav, 1996] surveys the multifarious ways in which the notion of context has been understood. Some think of contexts as being the same or similar to *situations* as understood, say, in situation theory [Barwise and Perry, 1983]. A situation is regarded as a collection of states of affairs, where states of affairs might be thought of as possible facts. An actual situation might even be identified with a particular spatio-temporal slice of the universe.

If contexts are thought of as situations, then it does make sense to regard the restriction of a class to those instances occurring in a particular context as a subclass of that class[1]. For on this construal of contexts, C2 is a subcontext of C1 if C2 is a restriction of C1 to some particular subset of the facts occurring in C1. Thus the set of instances of a class K that occur in context C2 will be a subset, and often a proper subset, of the set of instances of K occurring in C1. For example, if by default penguins don't fly and one considers, say, birds-inhabiting-Patagonia, which do fly by default, there's no reason to simply assume that penguins-inhabiting-Patagonia don't fly. (Perhaps unusual gravitational conditions in Patagonia give all birds the ability to fly.)

Another interpretation of "context" is as a state of belief. On this interpretation a subcontext represents an *extension* of its parent's state of belief. Thus a subcontext represents a state of belief in which all of the beliefs in the parent are still held plus other beliefs that are added in the subcontext (provided the new beliefs are consistent with the old ones).

On this interpretation of what a context is, defaults holding in supercontexts are inherited by a context unless overridden. Hence the context birds_can_fly simply adds to the information present in its parent context birds, the information that by default birds can fly. This default rule is consistent with the information contained in birds that by default penguins cannot fly and so that piece of information is inherited by birds_can_fly. The default inference rules for AKS class/context hierarchies can therefore be justified when contexts are understood as belief states[2].

Conclusion

We have described an implementation of a default reasoning system that combines class inheritance hierarchies with contexts. A translation of assertions in our system into default rules in Reiter's system of default logic was given. These default rules can be justified when the notion of a context is understood as a state of belief and a subcontext as an extension or revision of a state of belief.

References

Akman, V. and Surav, M. 1996. "Steps Toward Formalizing Context," in *AI Magazine* 17(3), pp. 55-72,

Barwise, J. and Perry, J. 1983. *Situations and Attitudes*. Cambridge:MIT Press.

Etherington, D. W., 1988. *Reasoning with Incomplete Information*. Los Altos: Morgan Kaufmann Publishers.

Minsky, M. 1975. "A framework for representing knowledge" in P. Winston ed., *The Psychology of Computer Vision*. New York: McGraw-Hill, pp. 211-280.

Reiter, R. 1980. "A logic for default reasoning" in *Artificial Intelligence 13*, North-Holland, pp. 81-132.

Shore, J. "Relative Entropy, Probabilistic Inference, and AI." In *Proceedings of the First Conference on Uncertainty in Artificial Intelligence*, ed. L. Kanal and J. Lemmer, Elsevier Science Publishing Co., Inc., 1985.

Touretzky, D. S., 1984. "Implicit ordering of defaults in inheritance systems." In *Proceedings of the Fifth National Conference on Artificial Intelligence*, pp. 322-325.

Acknowledgement

This material is based upon work supported by the Air Force Research Laboratory under Contract No. F30602-01-C-0041. Any opinions, findings and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the United States Air Force.

[1] In Cyc™, for example, it is assumed that quantifiers in rules for a given microtheory range only over objects existing in the situation described by the microtheory.

[2] A more formal argument for this conclusion can be given, using maximum relative entropy updating for belief revision and a probabilistic/statistical model of default rules.