Variables in Hypotheses

David Poole
Logic Programming and Artificial Intelligence Group,
Department of Computer Science,
University of Waterloo,
Waterloo, Ontario, Canada, N2L3G1
dlpoole@waterloo.csnet

Abstract

In many applications we want to build systems which must test the consistency of some theory (or set of axioms). This problem is general to many applications, for example abduction, learning, default reasoning, diagnosis, and is examined here in the context of theory formation from a fixed set of possible hypotheses [PGA87, Poole86]. There is a problem which arises when we are generating theories that contain variables. Two solutions are examined, the first where we are only allowed to have ground instances in theories formed, and the second where we may have universally quantified variables in the theory. It is shown that for the second case that the solution of reverse Skolemisation is not adequate to solve the problem, nor is any naive pattern matcher. A solution for both cases is outlined.

1 Introduction

We consider here the problem of checking the consistency of a theory generated by a program. We consider the problem in terms of a theory formation system which has a fixed set of possible hypotheses (i.e., we are assuming some other system is supplying the general forms we can to assume. This study is independent of how the possible hypotheses are generated). This is a problem because variables in a generated theory somehow need to have their quantification reversed when checking consistency. This paper shows that some proposed solutions do not work, and provides a solution to the problems where we don't allow variables in our hypotheses, and the general case where we allow arbitrary instances of defaults.

2 Formal Semantics

We use the standard syntax of the first order predicate calculus, with variables in upper case. We assume we have the following two sets of formulae:

F is a set of closed formulae (called *facts*), which we are given as true

Δ is a set of formulae, each instance of which can be used as a possible hypothesis

We say formula g is *explainable* if there is some D, a set of instances of elements of Δ, such that

$$F \cup D \models g$$

$$F \cup D \text{ is consistent}$$

D is said to be the theory that explains g.

Without loss of generality, we assume that g is variable free, and is an atom. We also assume that elements of Δ are atomic and do not contain bound variables. These assumptions do not restrict the expressiveness of the system, but make analysis simpler.

N.B. $w \in \Delta$ is equivalent to [Reiter80]'s normal default : Mw/w [Poole86]

3 Implementation

The obvious way to implement explainability [Reiter80, PGA87] is to note that both proving the observations, and testing consistency are the role of a theorem prover. Intuitively, the idea is to try to prove the goal from F and Δ, and make D the set of instances of Δ used in the proof. A complete theorem prover is an appropriate tool to check whether $F \cup D$ is consistent (by failing to prove inconsistency). Checking consistency corresponds to showing that the theory does not predict anything known to be false.

In this paper I assume that we are using some sort of complete resolution theorem prover (see eg. [Chang73]) to generate the instances of hypotheses which imply the goal. The results, however, do not seem to be restricted to such systems.

There is a problem which arises when there are variables in the D generated. Consider the following example:

Example 1 Let $\Delta = \{p(X)\}$. That is, any instance of p can be used if it is consistent. Let $F = \{\forall Y(p(Y) \Rightarrow g), \neg p(a)\}$, that is g is true if there is some Y for which $p(Y)$ is true.

g is explainable with the theory $\{p(b)\}$, which is consistent with F (consider the interpretation $I = \{\neg p(a), p(b)\}$ on the domain $\{a, b\}$), and implies g. So according to our semantics above, g is explainable.

However, if we try to prove g, we generate $D = \{p(Y)\}$ where Y is free (implicitly a universally quantified variable). The existence of the fact $\neg p(a)$ should not make it inconsistent, as we want g to be explainable.

Theorem 1 *In proving explainability, it is not adequate to only consider interpretations in the Herbrand universe of some set of formulae.*

Proof consider the example above; the Herbrand universe is just the set $\{a\}$. Within this domain there is no consistent theory to explain g. □

This shows that Herbrand's theorem is not applicable to the whole system. It is, however, applicable to each of the deduction steps [Chang73].

4 Ground Instances of Defaults

Consider first the case where we only allow ground instances of possible hypotheses in a theory. A ground instance is defined to be one without variables or Skolem constants.

The procedure[1] to compute explainability, when we only allow ground instances of defaults in theories, is

1. Skolemise F, forming F_{sk} (free variables are universally quantified);
2. try to prove g using elements of F_{sk} and Δ as axioms. Make D_0 the set of instances of Δ used in the proof;
3. reject D_0 if it contains a Skolem function (that is try to find another proof of g);
4. Form D_1 by replacing free variables in D_0 with unique constants;
5. add D_1 to F and try to prove an inconsistency. If complete search for a proof fails, g is explainable.

Example 2 consider F and Δ as in example 1 above. If we try to prove g, we use the hypothesis instance $p(Y)$. This means that g is provable from any instance of $p(Y)$. To show g cannot be explained, we must show that all of the instances are inconsistent. The above algorithm says we replace Y with a constant β. $p(\beta)$ is consistent with the facts, so that we can show g is explainable.

Let us first try to justify this procedure.

If g is explainable, there is a ground theory D which explains g. Some more general set D_0 of instances of defaults can be generated in the manner described above such that $D = D_0\theta$ for some θ (this is a direct corollary of the lifting lemma [Chang73, page 84]).

The third step of the procedure enforces the groundness of defaults found. The fourth and fifth steps follow from checking if $\exists \overline{X} D_0$ is consistent by Skolemising the $\overline{X}$ (the free variables in D_0).

5 Arbitrary Instances of Defaults

Sometimes we don't want to be restricted to just ground instances of defaults. Consider the following examples:

Example 3 Consider the blocks world, where we only want positive knowledge about which blocks are on each other, and we want the closed world assumption for "on". This is done by having the defaults: $\Delta = \{\neg on(X, Y)\}$.

If we have

$$\begin{aligned} F = \{ \ & \forall X((\neg\exists Y \ \mathrm{on}(Y, X)) \Rightarrow \mathrm{cleartop}(X)), \\ & \mathrm{on}(a, b)\} \end{aligned}$$

This says that a block has a clear top if there is nothing on it, and that block a is on block b. We want to conclude that b does not have a clear top, and all other blocks have a clear top.

Example 4 Let $\Delta = \{ontable(X)\}$. That is, we may assume that any block is on the table. Let

$$\begin{aligned} F = \{ \ & (\exists Y \ \mathrm{red}(Y) \wedge ontable(Y)) \Rightarrow g \\ & \exists X \ \mathrm{red}(X)\} \end{aligned}$$

We want to say that g is explainable as there is a red thing on the table, namely the object that we know is red (but do not know its name). The ground procedure would reject such an answer, as it must know the name of the individual said to exist.

We extend the definition of explainability to allow arbitrary instances of the possible hypotheses in our theories. In particular we want to be able to assume a default for all individuals we can.

If we want to expand the procedure given in the previous section, we have to consider how to handle

[1] This problem is, in general, undecidable; this procedure has the property that if it halts, it has computed a correct answer, and if a provable answer exists this (non-deterministic) procedure can compute it.

existentially quantified variables. The standard way to do this is to give names to the individuals said to exist (i.e. Skolemisation, [Chang73]). If we Skolemise it allows us to use normal resolution theorem provers. However, we must consider the Skolem functions in the theories generated.

It has been suggested [Bledsoe78, Cox84] that we "reverse Skolemise" the generated hypotheses. If we can prove their negation, we have shown the theory inconsistent; if a complete theorem prover fails to prove their negation the theory is consistent. This is equivalent to unifying the reverse Skolemised form with provably inconsistent instances of possible hypotheses.

Unfortunately, no such pattern matching program will work in general.

Theorem 2 *There can be no algorithm which does pattern matching on the instances which lead to the goal to be explained, and the instances which are inconsistent such that the goal is explained if and only if the pattern matcher fails.*

Proof: To prove this it is adequate to show two examples which have identical inconsistent hypotheses and syntactically identical instances which can prove the goal, but have opposite answers.

The examples we use are based on having just one simple default, namely that any individual is in the table: Consider $\Delta = \{ontable(X)\}$ and

$$\begin{aligned} F = \{\ & \neg\text{ontable}(a), \\ & \text{red}(a), \\ & (\forall X \ \text{ontable}(X)) \Rightarrow g_1, \\ & (\forall X \ \text{red}(X) \vee \text{ontable}(X)) \Rightarrow g_2\} \end{aligned}$$

That is, there is one block (a) that is not on the table, and is red. g_1 is explainable if everything is on the table. g_2 is explainable if all non-red things are on the table. According to our semantics g_1 should not be explainable (as we can't assume a is on the table), but g_2 should be explainable (assuming that everything except a is on the table).

When attempting to compute their explanations, we note that exactly the same instances of hypotheses lead to each goal, and exactly the same instances are inconsistent. Put into Skolem normal form this becomes:

$$\begin{aligned} F_{sk} = \{\ & \neg\text{ontable}(a), \\ & \text{red}(a), \\ & \text{ontable}(c_1) \Rightarrow g_1, \\ & \text{ontable}(c_2) \Rightarrow g_2, \\ & \text{red}(c_2) \Rightarrow g_2\} \end{aligned}$$

To prove each g_i we generate the theory $\{ontable(c_i)\}$, and the only inconsistent instance of hypotheses is $ontable(a)$. Note that the last clause is not used in either the proof of g_2 nor in proof of inconsistency. □

6 Building the Knowledge Base

The problem we have is that we have lost the context of what the Skolem constants represent. In this section we show how Hilbert's ε-symbol can be used to keep track of which functions the Skolem functions denote.

Hilbert's ε-symbol is a notational device to implicitly describe an individual said to exist. $\varepsilon x.P(x)$ means, intuitively "an x such that $P(x)$ is true". This was designed to eliminate existential variables through the equivalence:

$$\exists X \ w[X] \equiv w[\varepsilon X.w[X]]$$

where $w[X]$ is any well formed formula parameterised by X. See [Leisenring69] for a detailed description of Hilbert's ε-symbol.

When Skolemising (see [Chang73]), we replace

$$\forall X_1 ... \forall X_n \exists y \ w[X_1, ..., X_n, Y]$$

(where $w[X_1, ..., X_n, Y]$ is a well formed formula with free variables $X_1, ..., X_n, Y$) with

$$\forall X_1 ... \forall X_n \ w[X_1, ..., X_n, f(X_1, ..., X_n)]$$

We should also define what f is. We can use Hilbert's ε-symbol to define f:

$$f = \lambda X_1, ..., \lambda X_n.\varepsilon Y.w[X_1, ..., X_n, Y]$$

that is

$$f(X_1, ..., X_n) = \varepsilon Y.w[X_1, ..., X_n, Y]$$

To build the knowledge base, Skolemise all existentially quantified variables, and record the definitions of all Skolem functions and constants. In the Skolemised form all variables are universally quantified and so explicit quantification can be removed.

7 The General Explanation Procedure

The procedure outlined here is an extension of the one presented in section 4. See [Poole87] for more details.

We are trying to solve the problem of given some g, to decide whether it is explainable or not. That is, if there are assumptions from Δ which can be consistently added to F to imply g.

The following describes the algorithm:

Let F_{sk} be the Skolemised form of facts F. We assume we have recorded the definitions of all Skolem functions.

Try to prove g from F_{sk} and Δ. Let D_0 be the set of instances of elements of Δ used in the proof.

Let D_1 be a grounding of D_0. That is, we replace free variables in D_0 with unique constant symbols. We then know

$$F_{sk} \cup D_1 \models g$$

by construction (given our proof procedure is sound).

Form D_2 by replacing each instance of a Skolem function $f_i(t_1, ..., t_n)$ (where $f_i = \lambda v_1 ... \lambda v_n \varepsilon y.w_i[v_1, ..., v_n, y]$) in D_1 with a unique variable x_i. We know $(\forall x_1, ..., \forall x_m\ D_2[x_1, ..., x_m])$ implies D_1 and does not contain any Skolem functions, and so

$$F \cup (\forall x_1, ..., \forall x_m\ D_2[x_1, ..., x_m]) \models g$$

We now find out which instances of $D_2[x_1, ..., x_m]$ we cannot assume. To do this, we try to prove $\neg D_2[x_1, ..., x_m]$ from F_{sk}. We collect up the instances of the x_i proven [Green69].

Suppose we find the answer[2]:

$$\forall (x_1 = c_1 \wedge x_2 = c_2 \wedge ... \wedge x_m = c_m)$$

We know that we cannot assume any instance of $D_2[c_1, ..., c_m]$.

For those instances which we need in the above proof that we cannot use (because they are inconsistent), we must find an alternate explanation. The instances we need for the proof of g are $x_i = f_i(t_1, ..., t_n) = \varepsilon y.w_i[t_1, ..., t_n, y]$ The ones we must reexplain are where $x_i{=}c_i$ (for every instance of c_i). That is, for the case $w_i[t_1, ..., t_n, c_i]$ we must find an alternate proof to that found initially (for all instances of the c_i).

For each inconsistency of the above form found, we must try to reexplain g from the case

$$F' = F \wedge \exists \bigwedge_{i=1}^{m} w_i[t_1, ..., t_n, c_i]$$

If F' is consistent, then we can explain g only if we can explain g (using a different set of instances of elements of Δ) from F'. If F' is inconsistent, then inconsistency found is irrelevant to the theory needed to explain g.

In this description, we have ignored the problems of Skolem functions appearing in the c_i and the t_i, as well as disjunctive answers from the proof of $\neg D_2$. See [Poole87] for full details.

Example 5 Consider the blocks world of example 3. Let

$\Delta = \{\neg on(X, Y)\}$

$F = \{\ \forall X (\neg \exists Y \text{ on}(Y, X)) \Rightarrow \text{cleartop}(X),$
$\quad \text{on}(a, b),$
$\quad \text{cleartop}(b) \Rightarrow g_b,$
$\quad \text{cleartop}(c) \Rightarrow g_c\}$

That is, we can explain g_b if b has a clear top, and explain g_c if c has a clear top.

Skolemising the facts gives,

$F_{sk} = \{\ \neg \text{on}(f(X), X) \Rightarrow \text{cleartop}(X),$
$\quad \text{on}(a, b),$
$\quad \text{cleartop}(b) \Rightarrow g_b,$
$\quad \text{cleartop}(c) \Rightarrow g_c\}$

where $f(X) = \varepsilon Y. \neg \text{on}(Y, X) \Rightarrow \text{cleartop}(X)$.

We can prove g_b, generating the theory $D_0 = \{\neg on(f(b), b)\}$. $D_1 = D_0$. The corresponding $D_2 = \{\neg on(Y, b)\}$. We then try to generate all answers to $on(Y, b)$, which is provable ($Y = a$).

We then try to reexplain g_b making

$$a = f(b) = \varepsilon Y. \neg \text{on}(Y, b) \Rightarrow \text{cleartop}(b)$$

That is, we try to explain g_b (using an alternate proof to above) from

$$F \wedge \neg \text{on}(a, b) \Rightarrow \text{cleartop}(b)$$

which cannot be done.

We can explain g_c as we cannot prove the negation of the corresponding D_2. That is, we cannot prove $\exists Y\ on(Y, c)$.

Example 6 Consider the example in theorem 2 above. The Skolem constants, c_i have different definitions,

$$c_1 = \varepsilon X.\text{ontable}(X) \Rightarrow g_1$$

$$c_2 = \varepsilon X.(\text{red}(X) \vee \text{ontable}(X)) \Rightarrow g_2$$

In each case the generated theory is $D_0 = \{ontable(c_i)\}$. So for each of these we have $D_2 = \{ontable(X)\}$. We try to prove

$$\neg ontable(X)$$

This can be proven, in each case, for $X = a$.

In the first case, we then have to explain g_1 from

$$F \wedge (\text{ontable}(a) \Rightarrow g_1)$$

which cannot be done (using a different theory). Hence g_1 cannot be explained.

To prove the second case inconsistent, we have to explain g_2 from

$$F \wedge ((\text{red}(a) \vee \text{ontable}(a)) \Rightarrow g_2)$$

[2] If w is a formula, $\forall w$ is the universal closure of w. That is if w has free variables $v_1, ..., v_k$ then $\forall w$ is defined to be $\forall v_1 ... \forall v_k\ w$. Similarly $\exists w$ is defined to be the existential closure of w.

that is, we have to try to explain g_2 from

$$F \wedge (\mathrm{red}(a) \Rightarrow g_2) \wedge (\mathrm{ontable}(a)) \Rightarrow g_2)$$

This can be done (as red(a) is in F). Thus g_2 can be explained.

Example 7 Consider example 4. $\Delta = \{\mathit{ontable}(X)\}$.

$$\begin{aligned} F = \{ \; & (\exists Y \ \mathrm{red}(Y) \wedge \mathit{ontable}(Y)) \Rightarrow g \\ & \exists X \ \mathrm{red}(X)\} \end{aligned}$$

Skolemised, the facts become,

$$\begin{aligned} F_{sk} = \{ \; & (\mathrm{red}(Y) \wedge \mathit{ontable}(Y)) \Rightarrow g \\ & \mathrm{red}(c)\} \end{aligned}$$

where $c = \varepsilon X.\mathrm{red}(X)$. We can explain g, generating $D_0 = \{\mathit{ontable}(c)\}$ which gives $D_2 = \{\mathit{ontable}(X)\}$. We cannot prove $\neg \mathit{ontable}(X)$ for any X, so that g is explained.

Example 7A Let $F_1 = F \cup \{\mathit{red}(a), \neg \mathit{ontable}(a)\}$. g should not be explainable from F_1, as there is no reason to assume that there is another individual which is also red. We can prove, $\neg \mathit{ontable}(X)$ for $X = a$ and cannot explain g from $F \wedge \mathrm{red}(a)$.

Example 7B Let $F_2 = F \cup \{\neg \mathit{red}(a), \neg \mathit{ontable}(a)\}$. g should be explainable from F_2, as we know there is another individual which is red which we can assume it is on the table. We can explain g with the same theory. We can prove $\neg \mathit{ontable}(X)$ for $X = a$, but $F' = F \wedge \mathrm{red}(a)$ is inconsistent, and so the inconsistency is irrelevant to the theory needed.

8 Conclusion

There are many areas in which this problem arises. Some people have assumed that it is sufficient to consider the Herbrand Universe [Reiter80]. Others have tried to define a "reverse Skolemisation" algorithm which can be applied to the hypotheses generated, and unified with the instances leading to inconsistencies [Cox84, Bledsoe78]. We have shown that both of these ideas cannot work.

We have shown that we need to keep track of the context in which Skolem functions are defined, and have shown how this can be done by using Hilbert's ε-symbol. A procedure is outlined which solves this problem for the case of no Skolem functions in the inconsistencies found. [Poole87] gives the general solution to the problem.

Acknowledgements

This research was supported under NSERC grant A6260. Thanks to Eric Neufeld, Paul Van Arragon, Denis Gagné, Bruce Kirby and Randy Goebel for valuable discussions, arguments and comments on an earlier draft of this paper.

References

[Bledsoe78] Bledsoe,W.W. and Ballantyne,A.M., *Unskolemizing,* University of Texas at Austin, Math Dept Memo ATP-41A, July 1978.

[Cox84] Cox,P.T. and Pietrzykowski,T., "A Complete, Nonredundant Algorithm for Reverse Skolemisation", *Theoretical Computer Science, 28*, pp. 239-261.

[Chang73] C. Chang and R. Lee, *Symbolic Logic and Mechanical Theorem Proving*, Academic Press, 1973.

[Green69] C. Green, "Application of Theorem Proving to Problem Solving", *Proc. 1st International Joint Conference of Artificial Intelligence*, pp. 219-239.

[Leisenring69] A. C. Leisenring, *Mathematical Logic and Hilbert's ε-symbol*, MacDonald Technical and Scientific, London, 1969.

[Poole86] D. Poole, *Default Reasoning and Diagnosis as Theory Formation*, Technical Report CS-86-08, Department of Computer Science, University of Waterloo, March 1986, 19 pages.

[Poole87] D. Poole, *Building Consistent Theories*, Technical Report, Department of Computer Science, University of Waterloo, May 1987.

[PGA87] D. Poole, R. Goebel and R. Aleluinas, "Theorist: a logical reasoning system for defaults and diagnosis", in N.Cercone and G.McCalla (Eds.) *The Knowledge Frontier: Essays in the Representation of Knowledge*, Springer Varlag, New York, 1987, pp. 331-352.

[Reiter80] R. Reiter, "A Logic for Default Reasoning", *Artificial Intelligence*, Vol 13, pp. 81–132.