Integrating Definitions and Defaults

Bernhard Pfahringer*

22 September 1989

1 Introduction

This paper sketches a few ideas on how to integrate default reasoning into a KL-ONE-like language. These ideas will be presented and motivated intuitively via examples. To date they are neither worked out formally nor implemented, hence a decent classifier would recognize this paper as an instance of the generic concept research proposal.

2 Defaults in Knowledge Representation

In [4], Etherington describes two camps in knowledge representation dealing with defaults, namely inheritance networks and default theories. Both camps have yet to offer satisfying solutions to the problem of representing both universally true statements and typically true statements within one formalism. Default theories (see e.g.[3], which will also provide further pointers to literature) look very pleasing from a formal point of view, but cannot handle the complexity of non-toy examples due to the pitfalls of general theorem proving. Inheritance networks on the other hand are totally unable to represent contingent truth, as was shown by Brachman in [1]. At least the semantics of inheritance are better understood now due to research conducted by Touretzky [8], Etherington [4], or Horty et.al. [5], among others.

*Austrian Research Institute for AI, and Department for Medical Cybernetics and AI, University of Vienna
Recently, a third camp seems to materialize as some systems of the KL-ONE family of languages try to incorporate facilities for representing default knowledge:

- In [2], a possible P(prototype)Box for a KL-ONE-like language is described, but has not been implemented. For every TBox concept a prototype can be specified, possibly specializing role definitions or structural descriptions. Possible inferences illustrated are:

  1. Recognition of an ABox instance as being a typical instance of a generic concept,
  2. Propagation of prototypical knowledge to ABox instances,
  3. Identification of concepts as being typical for a certain superconcept, when the concept is identical to the super-concept’s prototype,
  4. Recognition of an ABox instance as being an instance of a primitive generic concept, because the instance matches the prototype of the primitive concept.

- SB-ONE [6] lets the user specify defaults for number and value restrictions of roles, a formal semantics for this syntax is being worked out currently.

- In [7], a more expressive D(efault)Box for LOOM is proposed, already providing limited support of default values and recognition of closed-world assumptions as well as sketching a non-monotonic classifier/recognizer. Its use seems to be restricted to the ABox, though.

3 A Primitive for Defaults

We think that it is possible to model most of the commonly known examples for default reasoning by introducing just one additional primitive into KL-ONE-like languages, which we would like to call :\textit{typically-implies}\textsuperscript{1}. Informally,

\[
C_1 :\text{typically-implies} C_2
\]

\textsuperscript{1}LOOM's \textit{implies} inspired this choice
expresses the knowledge that typically (in case no contradictory information is known) a sub-concept/instance of C1 will also be a sub-concept/instance of C2. In the TBox this fact can be represented the following way: the common sub-concept of C1 and C2, say C1-C2, is computed and a :typically-implies link connecting C1 to C1-C2 is established. The very same mechanism is used in LOOM to represent LOOM’s :implies primitive, and can easily be generalized to allow for more than one typically-implied concept.

Typically-implied concepts ensure maximal expressiveness, as via a concept definition any valid TBox expression - from the most simple to really complex ones - can act as defaults. There is, of course, potential for some form of syntactic sugaring easing the definition of simple, recurring schematas of default implications. To illustrate the use of our newly defined primitive, let us look at a classical example, namely the so-called “Nixon Diamond”:

1. Quakers are typically pacifists.
2. Republicans are typically not pacifists.
3. Nixon is both a quaker and a republican.

One possible representation of the Nixon Diamond is:

(defconcept Pacifist :primitive :disjoint NonPacifist)

(defconcept NonPacifist :primitive :disjoint Pacifist)

(defconcept Quaker :primitive :typically-implies Pacifist)

(defconcept Republican :primitive :typically-implies NonPacifist)

(definstance Nixon :isa (Quaker Republican) )

This should result in a TBox containing two additional concepts Pacifist-Quaker and NonPacifist-Republican being the typically-implied concepts of Quaker and Republican respectively. But what about the ABox instance Nixon? Obviously the recognizer cannot consistently assume it being an instance of both Pacifist-Quaker and NonPacifist-Republican, due to :disjoint markings of Pacifist and NonPacifist in the TBox; on the other hand there is no information available for choosing one of the two alternatives. Like
in [5] we argue for not applying any of the conflicting default implications in this case, and we will develop a modified version of their algorithm for skeptical inheritance to cope with situations like the Nixon Diamond and more complex ones.

Now this was all ABox reasoning, but will default implications have any impact on the TBox, too? We have to answer yes, as we would like to have default implications be propagated to sub-concepts in the TBox in roughly the same manner as sketched above for the ABox. Let us illustrate that with one an AI’s famous mammals, the elephant:

(defconcept Elephant :primitive :typically-implies 4LeggedElephant)

(defconcept 4LeggedElephant :isa Elephant :role (Legs :nr 4))

(defconcept RoyalElephant :isa Elephant :primitive)

We expect the system to create 4LeggedRoyalElephant as being the typically-implied concept of RoyalElephant. This is the default implication inherited from Elephant and it is not contradicted by neither a definitional component of RoyalElephant nor by any other applicable default information (as their is no other default in this simple setting). Introducing that funny 4LeggedRoyalElephant explicitly, represents the fact that instances of RoyalElephant will be assumed to be typically four-legged. The benefit gained by such an explicit representation seems twofold:

- Possibly time-consuming skeptical inheritance of all the default implications is computed only once and cached for later use, just like a classical TBox classifier tries to identify unstated subsumption relations between concepts and caches them.

- Consequences of the interplay between definitions and default implications can be displayed to and inspected by the knowledge base designer. That way the knowledge base designer can validate whether given definitions and defaults capture the intent or not, thus possibly leading to modifications of the knowledge base.

The Elephants can also illustrate the sound behaviour of our proposal when such pathological cases like the one described in [1] are encountered. Let us introduce:
(defconcept 3LeggedElephant :isa Elephant :role (Legs :nr 3))

3LeggedElephant will be just installed as a proper sub-concept of Elephant, and will not inherit any default implication, as this default information is conflicting with part of the definition of the new concept. Furthermore, 3LeggedElephant will never subsume 4LeggedRoyalElephant or vice versa, prohibiting the definition of such unhealthy cycles as are elaborated in [1].

4 Another Primitive blocking Defaults

Coping with the problem of conflicting defaults in multiple inheritance situations, a further primitive seems valuable, which allows for blocking inheritance of defaults of certain concepts. This primitive will be called not-a-typical. Returning to the Nixon Diamond we will show its usefulness. Assume we want to introduce the following concept for representing the type of persons like Nixon:

(defconcept NixonTypePerson :isa (Republican Quaker))

Like in the case of the ABox instance Nixon, the default implications contradict each other, therefore none is inherited according to the schema of skeptical inheritance. But what if we want to specify that NixonTypePersons are not typical Quakers and thus should not inherit defaults applicable to Quakers? A good and both aesthetically and formally pleasing solution could look like the following:

(defconcept NixonTypePerson :isa (Republican Quaker) :not-a-typical Quaker)

This will result in the creation of a NonPacifist-NixonTypePerson being the typically-implied concept of NixonTypePerson. Certainly, one could kludge work around the problem in this case by explicitly stating the defaults intended for NixonTypePerson, but there are situations where work-arounds just don’t exist. See also [8] for insights into the difference between blocking and explicitly overriding inherited defaults. This second new primitive somehow resembles NonApplicable clauses found in default theories.
5 Discussion

Comparing the proposal presented in this paper to those inferences proposed in [2] we find:

1. This inference is possible in principle, but what is the benefit of it? It might be useful for certain application programs.

2. These inferences are drawn via skeptical inheritance of defaults.

3. This inference is seemingly rendered obsolete, as no distinct PBox exists. But consider the following definition:

   (defconcept NixonTypePerson :isa (Republican Quaker Non-Pacifist))

An application program might be able to draw useful inferences from the fact that this differently defined *NixonTypePerson* is a sub-concept of *NonPacifist-Republican*, which in turn is the typically-implicated concept of *Republican*, but that the same line of reasoning does not hold for *Quaker*.

4. This seems to be a very interesting idea, as it would enhance the usefulness of primitive concepts in the recognition process, but possible consequences and pitfalls have to be examined carefully first.

For further insight into the value of our proposal, we will have to work out a formal semantics for the new primitives, to investigate the complexity of the informally described non-monotonic classification and recognition processes, and to build a reasonably efficient implementation. We will also have to investigate interaction of defaults in cases where a typically-impiled concept typically-implies further concepts, thus possibly leading to contradictory conclusions.

6 Acknowledgements

I am indebted to Christian Holzbaur for introducing me to \LaTeX, to Paolo Petta and Ernst Buchberger for proof-reading, to all my other colleagues for creating a very special working environment, and especially to my boss Robert Trapp for providing financial support.
References


