To appear in AAAI-84

The Tractability of Subsumption in Frame-Based Description Languages

Ronald J. Brachman
Hector J. Levesque

Fairchild Laboratory for Artificial Intelligence Research
4001 Miranda Avenue
Palo Alto, California 94304

ABSTRACT

A knowledge representation system provides an important service to the rest of a knowledge-based system: it computes automatically a set of inferences over the beliefs encoded within it. Given that the knowledge-based system relies on these inferences in the midst of its operation (*i.e.*, its diagnosis, planning, or whatever), their computational tractability is an important concern. Here we present evidence as to how the cost of computing one kind of inference is directly related to the expressiveness of the representation language. As it turns out, this cost is perilously sensitive to small changes in the representation language. Even a seemingly simple frame-based description language can pose intractable computational obstacles.

1. Introduction

There are many different styles of knowledge representation system in use in Artificial Intelligence programs, but they all have at least this in common: the representation system is supposed to provide both a repository for the beliefs of the knowledge-based system in which it is embedded, as well as automatic inferences over those beliefs. Typical inferences automatically computed by AI representation systems include inheritance of properties, set membership and set inclusion, part/subpart inferences, type subsumption, and resolution.

Here we address a fundamental problem in the nature of the service to be provided by knowledge representation systems: the greater the expressiveness of the language for representing knowledge, the harder it becomes to compute the needed inferences (see [7] for an overview of this tradeoff). In this brief paper, we present a formal analysis of the computational cost of expressiveness in a simple frame-based description language. We illustrate how great care needs to be taken in the design of a representational facility, even when our intuitions about the language tell us that it is a simple one. As it turns out, even an apparently modest representation language can prove intractable.

2. Subsumption in Frame Languages

Among the more popular representation languages in use today are those based on the notion of *frames* (see, for example, [1], [3], and [9]). Frames give us the ability to define structured types; typically a frame comprises a set of more general frames (its *superframes*) as well as a set of descriptions of the attributes (*slots*) of instances of the frame. The most common type of slot description specifies a restriction on the value of the filler of the slot for all instances of the frame. The restriction can be as specific as a particular value that all instances of the frame must exhibit (alternatively, the value may be just a *default*, in which case an individual inherits the value provided he does not override it), or it may be a more general constraint on attribute values, in which case this *value restriction* is usually a pointer to another frame. Less commonly, the number of required fillers is also specified in a slot restriction (often in terms of a minimum and a maximum number of attribute values).[1] The generalization relation between frame and superframe, or between two frames where one is simply a more restricted version of another, implicitly forms a *taxonomy*, or *inheritance hierarchy*.

Notationally, a frame might be defined by a list of superframes (with either an explicit or implicit "*isa*" relation [2]), followed by a set of slot restrictions expressed by attribute/value-description pairs (with attribute and value-description usually separated by a colon). For example, the simple frame

```
[PERSON
    child (≥ 1):
    son:          LAWYER
    daughter:     DOCTOR]
```

is intended to be a structured type representing the concept of a person that has at least one child, and all of whose sons (*i.e.*, male children) are lawyers and all of whose daughters are doctors. Similarly, the more complicated frame,

```
[STUDENT, FEMALE
    department:  COMPUTER-SCIENCE
    enrolled-course (≥ 3) :
            [GRADUATE-COURSE
                department:  ENGINEERING-DEPARTMENT]]
```

is intended to be a structured type that describes female Computer Science students taking at least three graduate courses in a department within a school of Engineering.[2]

There is a natural correspondence between this frame form of description and noun phrases in natural language. For example, the above frame might just as well have been written as "student and a female whose department is computer-science, and who has at least 3 enrolled-courses, each of which is a graduate-course whose department is an engineering-department." A simple set of translation rules would allow us to move easily from frame form to (almost) readable English.[3]

---

[1] While the use of number restrictions is not widespread, they have been used extensively in KL-ONE [3] and languages like it [4]. They seem to be a useful generalization of the existential reading of slots (see below), so we include them here.

[2] Typically, frames are given *names* as well; for example, we might have labeled our first example frame, "PROUD-PARENT". We have explicitly chosen to avoid these here so as to eliminate confusion about their meanings. For this paper, we are interested in relations among frames implied by their *structure* only (see below), and will assume that atoms are all independent.

[3] For example, the list of superframes would translate into a conjunction of nouns ("student and a female"). A slot that had only one filler might translate into a simple "whose" clause ("whose department is computer science"). And a slot