Reset reproduction of article published in *Artificial Intelligence*, Vol. 48, No. 3 (April 1991), pp. 261–297. Received September 1989, revised June 1990. Reprinted July 1994.

Two Theses of Knowledge Representation
Language Restrictions, Taxonomic Classification, and the Utility of Representation Services

Jon Doyle
Laboratory for Computer Science, Massachusetts Institute of Technology
545 Technology Square, Cambridge, Massachusetts 02139, USA

Ramesh S. Patil
USC Information Sciences Institute, 4676 Admiralty Way
Marina del Rey, CA 90292-6695, USA

Abstract

Levesque and Brachman argue that in order to provide timely and correct responses in the most critical applications, general purpose knowledge representation systems should restrict their languages by omitting constructs which require non-polynomial worst-case response times for sound and complete classification. They also separate terminological and assertional knowledge, and restrict classification to purely terminological information. We demonstrate that restricting the terminological language and classifier in these ways limits these "general-purpose" facilities so severely that they are no longer generally applicable. We argue that logical soundness, completeness, and worst-case complexity are inadequate measures for evaluating the utility of representation services, and that this evaluation should employ the broader notions of utility and rationality found in decision theory. We suggest that general purpose representation services should provide fully expressive languages, classification over relevant contingent information, "approximate" forms of classification involving defaults, and rational management of inference tools.

1 Introduction

Much recent work on general-purpose knowledge representation systems views knowledge bases as propositional databases which can answer some questions by making inferences from the stored propositions as well as simply retrieving stored propositions (see, for example, [11, 17, 18]). Abstractly, this means that knowledge representation systems are like theorem-proving programs, where telling the system some fact adds an axiom (more generally, changes the axioms so that the added fact is a consequence of the database propositions), and where answering a query reduces to deductions over the database propositions. Viewed this way, it is natural to require knowledge base retrieval to be sound (answers are always entailed by the axioms) and complete (all entailed propositions can be computed).

Classification of concept definitions with respect to taxonomic hierarchies is a good example of a logical knowledge base operation. Virtually all modern representation systems, beginning with semantic nets and frames, and continuing with theories of inheritance, organize their representations around taxonomic hierarchies. Taxonomic information pervades common and specialized knowledge and is used to organize knowledge and inference in virtually every field (see section 1.3). Classification is a fundamental means of taxonomic reasoning in which each newly defined concept is compared with previously encountered concepts and placed in the taxonomic hierarchy above the concepts it subsumes (because their definitions are more specific) and below those that subsume it (because their definitions are more general). Classification is also very general, as it is essentially equivalent to logical entailment, and so capable of supporting most forms of logical inference.

1.1 Language restrictions and restricted classification

Brachman, Levesque, and Fikes [6] argue that the inferential operations of a knowledge representation system should be efficient in addition to being sound and complete. They suggest viewing representation systems as embedded utilities or subsystems providing computational services in support of more general reasoning or problem solving. They argue that this role requires that the services it provides be dependable, that is, sound and operating efficiently in predictable (e.g., bounded) amounts of time. Because the knowledge base is supposed to be general purpose, efficiency has been interpreted to mean that the utility's worst-case time requirements should be small enough to allow adequate response to the most critical applications (and so be adequate for less critical applications as well). In particular, "adequate response" has been taken to mean time polynomial in the size of the knowledge base.[1] This need has motivated research to identify inferential operations that can be mechanized efficiently. However, general logical inference operations like classification are not always efficiently mechanizable. Indeed, Levesque and Brachman [19] have shown that there is a tradeoff between expressive power and the computational cost of classification which, depending on the richness of the language in which definitions are expressed, can range from tractable to undecidable.

To address this tradeoff, Levesque and Brachman recommend restricting the languages of general purpose knowledge representation by omitting constructs which would make classification (as the most expensive inferential service these systems provide) inefficient. They arrive at this approach after dismissing several alternative approaches for achieving adequate speed in inferential operations. They dismiss the possibility of speeding up computation through the use of distributed or parallel computers since such computers may improve the average efficiency of inferential operations but cannot guarantee that the worst-case costs do not exceed the specified limits. Similarly, they dismiss the possibility of obtaining efficient average responses because mere average efficiency is not adequate for the most critical, life-or-death applications. They also dismiss relaxing the requirements of soundness and completeness. Permitting classification to be unsound would undercut the assumed meaning of the answers returned. If the results of classification may be incomplete, definitions need not mean the same thing to the system that they do to the user. Either of these lapses from correctness can render answers undependable in critical applications. With these alternatives dismissed, Levesque and Brachman recommend the remaining path of restricting the language.

In summary, Levesque and Brachman [19, pp. 81-82] argue that

[1] This sense of "adequate response" is very generous, in that one can expect that the knowledge bases of broadly intelligent systems will be very large, so that even time linear in the size of the knowledge base could be significant.

1. General dependability requires that the inferences supplied by a knowledge base must be sound and complete. In particular, the system should be able to classify definitions soundly and completely.

2. General purpose systems must be quick enough for the most critical applications.

3. Classification's efficiency should dictate the design of languages because classification is the most expensive operation most KB's provide and because classification reduces to full inference.

4. Therefore, general purpose knowledge representation systems should restrict their languages by omitting constructs which require non-polynomial (or otherwise unacceptably long) worst-case response times for correct classification of concepts.

We call thesis (4) the *restricted language thesis*.

Motivated in part by the semantically confusing nature of many early representation systems, in which it was sometimes difficult to tell whether representations were meant to define prototypes or to assert facts, Brachman, Fikes and Levesque [6] draw on the logical distinction between terms and axioms to divide knowledge bases into two parts. The terminological knowledge base (TBox) stores terms and their definitions, while the assertional knowledge base (ABox) stores sentences constructed using these terms. The definitions stored in the TBox are taken as describing Platonic concepts and their analytic interrelations, while the assertions stored in the ABox state contingent facts about what is true of the world. The TBox and ABox each have their own language. These are called the terminological language and the assertional language, respectively. Brachman and Levesque [7, 4] argue that these two components should be separately designed and optimized for their respective tasks. In particular, they insist on a strict separation of the inferential mechanisms each employs, in that neither component can change or manipulate the information contained in the other. Inference in the TBox should not make any use of information in the ABox, and the ABox should only use terms in the TBox as non-logical symbols, relying on the TBox as a subroutine for all inferences relating terms. Specifically, they insist that taxonomic classification be confined solely to definitional information in the TBox. To summarize,

1. Definitions define terms.

2. Assertions use terms to state propositions.

3. Storing definitions and assertions separately clarifies the meanings of representations.

4. Optimizing definitional and assertional inferences separately maximizes the efficiency of the representation system.

5. Therefore, representation systems should restrict taxonomic classification to terminological definitions alone.

We call thesis (5) the *restricted classification thesis*.

These theses were promulgated in the context of the descendants of KL-ONE [8, 37], namely NIKL [41, 26, 15] (the New Implementation of KL-ONE), KL-TWO [41], KANDOR [30], KRYPTON [6], and BACK [42, 27]. While KL-ONE itself was designed earlier, without concern for language restrictions, the designs of KRYPTON, KANDOR, and BACK involved theoretical analyses (e.g., [19] and [27]) to see what linguistic constructs could be efficiently classified and what constructs should be included or forbidden. NIKL exhibits similar restrictions, but these came about by abandoning some poorly defined KL-ONE constructs rather than by explicitly attempting to ensure classificatory tractability.[2] As a group, these systems have been very influential. Their designers intended them to be general-purpose representational services, and they have been used in several practical applications.

1.2 Sketch of the counterargument

Levesque and Brachman's arguments have much to recommend them, and the classification-centered design for representation systems has proven remarkably fruitful in illuminating and demonstrating representational issues. Nevertheless, in this paper, we take the position that *the restricted language and restricted classification theses and their underlying assumptions are flawed*.[3] In wider practice the terminological facilities of such systems are so expressively impoverished that the very purpose set out for general purpose representational utilities is defeated. To summarize our arguments:

- Restricting languages by omitting constructs which require non-polynomial (or otherwise unacceptably long) worst-case response times for correct classification of concepts destroys the generality of the language and system.

  We present a list of representative examples that shows many important classes of definable concepts are inexpressible in the restricted languages of KL-ONE's descendants. These restrictions severely impair the utility of the system for developing applications. Language restrictions impose large costs in working around restrictions, when this is possible. When this is not possible, users must invent mechanisms external to the representation system for expressing their knowledge. In addition to reintroducing intractability concerns, this lack of standards for expression results in different ad hoc extensions to the language for each application. These gaps in expressive power thus mean that the restricted languages are not general purpose, contrary to the intent of their designers.

- Definitions inexpressible due to language restrictions must be entered as primitive concepts, which are unclassifiable, and which reduce the utility of classification.

  There are a large number of concepts that are unclassifiable by virtue of being natural kinds. The problem is exacerbated by a large number of "fake" primitives, concepts which are primitive only because their definitions cannot be expressed in the restricted language. Since these reduce the utility of classification, using classification's efficiency as the design criterion misplaces emphasis.

[2] We understand from Peter Patel-Schneider (personal communication, 1988) that NIKL was designed around an existing fast but partial KL-ONE classifier algorithm rather than through an initial theoretical analysis.

[3] The two theses of knowledge representation we criticize in some ways parallel the two dogmas of empiricism discussed by Quine [34], hence the title of this paper. Quine was critical of the distinction between analytic and synthetic truths and of the reductionist view that all concepts can be defined as logical constructs over elementary sense data.

- Restricting classification to purely definitional information significantly reduces its utility in practical applications.

  For some purposes, the most useful classifications take into account contingent information in addition to definitional information, as when persistent or slowly changing background knowledge determines the proper classification. For example, classification of diseases as treatable or not depends on the (slowly) evolving state of medical knowledge.

- Language restrictions which omit constructs entirely make more drastic limitations than are necessary to achieve the desired efficiency.

  For example, sparing use of problematic constructs need not increase asymptotic complexity. This is still a restriction, but not as drastic a one as forbidding all use of the constructs.

- Language restrictions aimed at ensuring the completeness of efficient classification prevents storage or retrieval of available information prior to full inferential understanding of this information.

  Completeness amounts to understanding all the consequences of concept definitions. Requiring completeness rules out the use of definitions that cannot be classified efficiently, for if they are permitted, the classifier must be incomplete. Thus requiring completeness prevents representation services from supporting imperfect and developmental reasoning and learning, in which there is no alternative to storing and retrieving information that is not yet completely understood.

- Requiring that general purpose systems be quick enough for the most critical applications misconceives the notion of runtime efficiency.

  This requirement is ill-defined, since for any specific level of efficiency one can always find some task which requires faster response than can be guaranteed. In addition, the worst case is not the general case; response time is not the only cost involved; and classification is not the only operation influencing the system's efficiency, so that optimizing classification alone may make the overall efficiency decrease. But more fundamentally, focusing on the costs of inference ignores consideration of the benefits they provide, or more generally, of the inferences' utility (in the sense of decision theory). Computational costs (whether time or otherwise) are not themselves measures of utility since each circumstance of application may involve different limits or costs to different amounts of time or space usage and different values for the information received.

Our arguments grow out of considerable experience with the inadequacies of NIKL (see [13]), but are not limited to NIKL. Recognition of the disappointing limitations of these systems occurred only through this experience and we do not believe these failures could have been predicted with certainty when the choices underlying these designs were made.

As an alternative, we argue that general purpose knowledge representation systems should provide

- Fully expressive languages,

- Tolerance of incomplete classification,
- Terminological classification over relevant contingent as well as definitional information,
- Nondeductive as well as deductive forms of recognition which permit "approximate" classification and "classification" of concepts involving defaults, and
- Rational management of inference tools.

More abstractly, our proposal means that logical soundness, completeness, and worst-case time complexity are not the right measures for evaluating the value of the services provided to the user or to the larger reasoning system making use of the knowledge base. Instead, a better view is that the purpose of representational services is to provide the rational or optimal conclusions rather than the logically sound conclusions (see also Doyle [9]), and that representation systems should be rational agents using the user's knowledge and purposes to cooperate with the user to accomplish those purposes.

We present the detailed arguments for each of the criticisms and alternative proposals after first setting out the fundamentals of taxonomic classification and the representational constructs of KL-ONE and its descendants.

1.3 Classification and its role in reasoning

Classification is based on the notion of subsumption of concepts. For the purpose of this discussion we take concepts to be unary predicates defined in a logical language. The *extension* of some concept in some model is the set of all individuals it describes in that model. When C and C' are concepts such that in every model the extension of C is a superset of the extension of C' (that is, every instance of C' is an instance of C), we say that C subsumes C'. Since subsumption mirrors the superset relation among concept extensions, the subsumption relation is reflexive, antisymmetric, and transitive, and hence a partial order.

Subsumption is closely related to the logical notion of entailment. Indeed, if a concept both subsumes and is subsumed by another concept, then its definition is logically equivalent to the definition of the other. Whether one concept subsumes another follows purely from their definitions. As noted earlier, subsumption is not decidable for sufficiently expressive languages. Restricted classes of languages do exist, however, for which subsumption is decidable, even in polynomial time. (See [19] for examples drawn from knowledge representation languages.)

To classify a new concept with respect to a hierarchy of concept definitions means to determine its place in the subsumption partial order. Thus as each new concept is defined, KL-ONE-style representation languages automatically compare its definition with previously encountered definitions and place the concept in the taxonomic hierarchy, explicitly recording the derived subsumption relationships. Equivalent definitions are collapsed so that they are represented by the same concept. In addition, KL-ONE-style systems use classification for answering queries. Whenever a description is used in a query, it is classified as though it were a new definition. Once this is done, the query can be answered from the recorded subsumption relation, which indicates where to look for inheritable information. For example, a request for the area of a three-sided polygon would first classify "three-sided polygon" as "triangle" and then use the formula associated with "triangle" for computing the area.

Of course, not all concepts can be classified in this way since not all concepts are explicitly defined. Specifically, *primitive* concepts are undefined terms used to denote incompletely described or undefinable concepts, concepts without expressed or expressible necessary and sufficient conditions. Primitives affect the operation of classification because the very meaning of "primitive" entails that two primitive concepts cannot be classified with respect to each other. Moreover, a defined concept cannot be classified as a subconcept of a primitive unless its definition explicitly contains the primitive. For example, assuming that "cat" is a primitive concept, an exact description of a cat cannot be classified as a cat unless the property "cat" is included in the list of its features. Placement of a concept under a primitive is thus fundamentally an act of assertion, as it states a relationship that cannot be derived from its definition.

Classification performs several important functions for knowledge-base maintenance. It detects redundancies by recognizing definitions equivalent to ones already present in the hierarchy. This allows the system to combine any information recorded separately about the equivalent concepts. Indeed, in KL-ONE-style systems one cannot define a concept to be distinct from others without either declaring it to be primitive or specifying additional elements of its definition which distinguish it from its superiors.

Classification also helps keep the hierarchy coherent and nontrivial by recognizing inconsistent and vacuous definitions. Any concept subsumed by the empty concept is equivalent to the empty concept (hence its definition is inconsistent), and any concept subsuming the universal concept is equivalent to the universal concept (hence its definition is vacuous). Classification also helps speed up query answering when the subsumption relation among concepts is explicitly and uniformly recorded whenever a new concept is added to the hierarchy. Precomputing all possible subsumption relations in this way simplifies retrieval algorithms by ensuring that if A subsumes B, the hierarchy contains an explicit path from A to B.

Another important motivation for automatic classification is that classification can be used to recognize concepts from their features. For example, if a human parent is defined to be a human who has a child, recognizing that some person is a parent allows one to apply reasoning appropriate to parents in addition to that appropriate merely to humans. More formally, Brachman and Schmolze [8, pp. 178,189] present an extended example of classification-based recognition in a language-understanding task. The prominence of concept-recognition tasks in Brachman and Schmolze's arguments and examples suggests that they viewed recognizing the essential nature of a situation or an object and acting accordingly as one of the central operations of reasoning. Similarly, the attention they pay to classification-based recognition suggests that they expected it to suffice for most cases of recognition. Others appear to have shared this expectation as well, for classification has been exploited for recognition tasks in the CONSUL system [20] and SUDO-PLANNER [43].

1.4 KL-ONE-style expressive constructs

As further background to our discussion, we outline the expressive constructs common to restricted KL-ONE-style languages. Readers familiar with these languages may wish to skip to Section 2. Briefly, the principal linguistic objects expressible in KL-ONE and its descendants are *concepts* and *roles*. Concepts express classes (unary predicates), and roles express relationships between classes (binary relations).

Each concept is constructed either by conjoining two or more concepts or by specializing a concept by adding restrictions to its description. KL-ONE provides three methods for specializing concepts: adding roles, which relate instances of the concept to instances of other concepts, adding restrictions to roles, and adding *structural descriptions*, which express the interrelations among the various roles of a concept and their potential fillers. These additional restrictions indicate the concept's internal structure, and play a key role in determining the meaning of the concept. Specifically, each concept's meaning is a function of the meanings of the more general concepts from which it is derived and the function by which these meanings are combined is determined by the concept's internal structure.

Structural interrelationships between various components of a concept's description were the quintessential elements of the SI-NET (Structured Inheritance NETwork) paradigm on which KL-ONE is based [2]. SI-NET and KL-ONE provided powerful languages for expressing these interrelationships, including a complete set of boolean connectives, quantification, set inequality, mapping functions, and a general "parametric individual" mechanism which allowed any concept defined in the knowledge base to be used as an n-ary predicate in specifying constraints among roles.[4] As these constructs suggest, KL-ONE was intended to provide a broad expressive coverage within an object- or concept-centered framework, not to restrict what could be defined.

KL-ONE's descendants, in contrast, all limit concept definitions in essentially the same way, namely to a common subset of KL-ONE's constructs which severely restricts the use of structural descriptions. For concreteness, we will use NIKL's nomenclature (but not its syntax) for these definitional constructs [41, 26, 15].[5] Some of the constructs in this subset are fully supported by the classifier and some are not. We discuss these in turn.

Fully supported constructs: The principal constructions for defining concepts are `CMeet` for making conjunctive definitions, `CMin` and `CMax` for placing number restrictions on the values of role fillers, `CRestrict` for placing type restrictions on role fillers, and `VRDiff`

[4]The structural interrelationships in KL-ONE were called *role set relations* (RSR's), which come in two types, the general notion of *structural descriptors* (SD's) and the much weaker notion of *role value maps* (RVM's). SD's are essentially full logical theories (using all logical connectives, quantifiers, and relations of arbitrary arity) which describe the interrelations of a concept's components, while RVM's express simple relationships of identity or inclusion between the sets of fillers of two roles (not necessarily of the same concept).

RSR's were never fully implemented in KL-ONE, and they were greatly weakened in the restricted language systems that followed. Brachman (personal communication, 1989) suggests that there were several reasons (including conceptual and computational difficulties and limited resources for development) why RSR's were eliminated or weakened in these later developments. But whatever the reasons, abandoning these language features fits right in with the restricted language approach to making classification efficient, as general SD's might express arbitrary logical relationships, and so not be amenable to efficient (or even decidable) classification.

[5]Details of the NIKL knowledge language and its functional interface can be found in the NIKL manual [36]. NIKL is unlike KRYPTON and KL-TWO in that it has no facility for making assertions that use the concepts it defines, but this will not matter for our discussion. NIKL also provides no adequate mechanism for representing instances of concepts.

Construct	*Use* *Meaning*
(`CMeet` C_1 C_2)	(`CMeet` PARENT MALE) $\lambda x.\mathrm{PARENT}(x) \wedge \mathrm{MALE}(x)$
(`CMin` R n)	(`CMeet` PARENT (`CMin` OFFSPRING 2)) $\lambda x.\mathrm{PARENT}(x) \wedge \lvert\{y : \mathrm{OFFSPRING}(x,y)\}\rvert \geq 2$
(`CRestrict` R C)	(`CMeet` PARENT (`CRestrict` OFFSPRING MALE)) $\lambda x.\mathrm{PARENT}(x) \wedge \forall y.\mathrm{OFFSPRING}(x,y) \supset \mathrm{MALE}(y)$
(`VRDiff` R C)	(`CMeet` PARENT (`CMin` (`VRDiff` OFFSPRING MALE) 1)) $\lambda x.\mathrm{PARENT}(x) \wedge \lvert\{y : \mathrm{OFFSPRING}(x,y) \wedge \mathrm{MALE}(y)\}\rvert \geq 1$
(R_1 R_2)	(OFFSPRING OFFSPRING) $\lambda x.\{y : \exists z.\mathrm{OFFSPRING}(x,z) \wedge \mathrm{OFFSPRING}(z,y)\}$
($\subseteq$ RC_1 RC_2)	($\subseteq$ MOTHER PARENTS) $\lambda x.\forall y.\mathrm{MOTHER}(x,y) \supset \mathrm{PARENTS}(x,y)$

Figure 1: Examples of the use and meanings of NIKL constructs.

for differentiating roles according to the types of fillers. Let C, C_1, C_2 be concepts and R be a role. Then (`CMeet` C_1 C_2) means the concept which conjoins C_1 and C_2. For example, (`CMeet` PARENT MALE) corresponds to the predicate $\lambda x.\mathrm{PARENT}(x) \wedge \mathrm{MALE}(x)$, which holds true of everything for which both PARENT and MALE hold true. (See Fig. 1.) (`CMin` R n) means that the concept is related to at least n distinct entities by the relation R. For example, (`CMeet` PARENT (`CMin` OFFSPRING 2)) represents the concept of a parent with at least two offspring. (`CRestrict` R C) means the concept for which each entity to which it is related via R is also an instance of C. For example, (`CMeet` PARENT (`CRestrict` OFFSPRING MALE)) represents the concept of a parent whose offspring are all male. Finally, (`VRDiff` R C) means the creation of a new role by restricting the range of R to concept C. For example, (`CMeet` PARENT (`CMin` (`VRDiff` OFFSPRING MALE) 1)) represents the concept of a parent with at least one male offspring.

Two constructions are provided for defining structural relations among roles of a concept, role chains (RC's) and role value maps (RVM's). Role chains designate compositions of role relations, so that the role chain $R_1 R_2$ designates the set of fillers of role R_2 of R_1 of the concept to which the role chain is attached. For example, the set of grandchildren of a person can be referred to using the role chain (OFFSPRING OFFSPRING) attached to the the concept of person. RVM's can be used to place a set of inclusion constraints among two sets of role fillers. For example, the RVM ($\subseteq$ RC_1 RC_2) states for each instance of the concept the set of fillers of RC_1 is a subset of the set of fillers of RC_2. Thus ($\subseteq$ MOTHER PARENTS) constrains the filler of the mother role of a concept to be subset of the fillers of its parent role. RVM's can be viewed as *asserting* subsumption relationships among some *unnamed* concepts, namely the concepts of the fillers of the two role chains. Note that

RVM's can also be used to equate the values or fillers of different roles in those situations in which each role or role chain mentioned in the RVM has unique fillers. This provides a limited mechanism for creating co-referential terms.

Partially supported constructs: In addition to the fully supported constructs, NIKL and other descendants of KL-ONE provide some expressive constructs whose meaning is not fully honored by the way the classifier works in these systems.[6] For example, NIKL also allows the user to declare a subset of specializations of some concept to be `Disjoint` (i.e., mutually exclusive) or to `Cover` the concept (i.e., to be exhaustive). It uses disjointness declarations to determine the consistency of concept descriptions, but it does not use either of these declarations in classifying concepts. Similarly, NIKL provides `Data` and `IData` data-structures which allow the user to attach to a concept arbitrary data which are local or inherited, respectively. These attached data declarations are also ignored when classifying concepts.

Being ignored by the classifier has grave consequences for the usefulness of these constructs in actually defining concepts. Specifically, though these constructs appear to confer significant expressive power, they are deceptive in that one cannot actually use them to mean what they seem to say. Because the classifier ignores the distinctions these constructs express, two concepts which differ only through one of these constructs will be identified as the same concept. The classifier will collapse them into a single concept and merge their descriptions. This means that further attempts to differentiate these two concepts are not possible, as they are no longer distinct. For example, the `Disjoint` and `Cover` constructs in principle allow us to "express" partitions, and so negations. Consider the concept EMOTIONAL-STATE and its two specializations HAPPY and UNHAPPY. For the purposes of this example, assume that the concept HAPPY is defined to be a primitive, and its negation UNHAPPY is defined using the `Disjoint` and `Cover` constructs, i.e., they are disjoint and taken together cover the parent concept of EMOTIONAL-STATE. However, since the classifier ignores the `Disjoint` and `Cover` annotations, UNHAPPY will be viewed as being indistinguishable from the definition of EMOTIONAL-STATE and merged with it. In short, we can not *define* UNHAPPY to be the negation of HAPPY because the negation expressed through these annotations is not a term-forming operation.[7]

As stated earlier, merging equivalent definitions (and so recognizing redundant definitions) is a motivating function of classification. But this same function renders the partially supported constructs ineffective in expressing definitions. Thus we ignore these constructs in assessing the utility of KL-ONE-style languages.

2 Language restrictions and generality

Restrictions on the concept-definition language naturally limit the expressive power of the representation system, and our first task is to see whether these limitations are significant in practice. Toward this end, we employ ordinary logical languages as the standard against

[6] This goes against the intent of Levesque and Brachman [19] that everything expressible has effect, that none of the expressions are treated as comments by the system. See section 6 for more discussion.

[7] Recognizing the difficulties posed by disjointness declarations, BACK permits disjointness to be declared only for primitive concepts (see [42, p. 32]).

which we compare the expressiveness of the limited languages. Most of the concepts we define require only first-order logic, though some are most simply and naturally expressed in second-order logic.

We employ logical languages for purposes of illustration only. Richer logics that permit expression of even more concepts would serve our arguments just as well.[8] The benefit we seek in using logic as an analytical tool is that it facilitates clear statement and analysis of representation problems, as in McCarthy and Hayes' [22] epistemological approach and Newell's [29] knowledge level analysis. But using it as an analytical tool does not mean that we advocate some particular logic as the proper choice of representation language, for other considerations may keep it from being the best choice for a practical representation language.[9]

Our task is assessing the effects of restricting terminological languages. In doing so, we may ignore statements expressed in the assertional languages because these do not form terms (concepts) in the taxonomic hierarchy, even though the assertional languages are typically full first-order languages with rich expressive powers. We say a concept is *definable* whenever it can be defined in logic, and we say it is *expressible* with respect to some language whenever it can be defined in that language. For the terminological languages of interest, every expressible concept is definable, but not every definable concept need be expressible. To assess the impact of language restrictions, we must judge the importance of concepts which are definable but inexpressible in these terminological languages.

2.1 Examples of inexpressible definitions

We now show that the restricted KL-ONE-style languages cannot express numerous common and important concepts that pervade the core of human knowledge, even though these concepts admit natural and simple definitions in logic. We demonstrate this by means of examples.[10] These examples fall into several broad categories: concepts defined using disjunction; negation and relative complement; conditionals; equivalence; particularization; recursive definition and transitive relations; functions over ordered sets; mappings between ordered sets; functions of sets and functions; and binary functions and n-ary relations. Each of these categories covers a large class of similarly defined and similarly important concepts and reflects a simple logical or mathematical notion. We expect that there are many more examples to be found either by attempting to model everyday knowledge and domains or by looking to more complex mathematical concepts (such as semigroups, groups, and bundles or other function spaces, because these have proven very useful for describing worldly objects or circumstances). The primary examples below are presented with both informal English statements and formal expressions in logic.

[8] In particular, we do not mean to imply that anything that can not be formally defined in logic is a natural kind or necessarily undefinable concept.

[9] Indeed, that is why KL-ONE-style languages do not simply use some subset of standard logical languages. See also McAllester et al. [21], who advocate one candidate for a practical logical representation language based on "taxonomic syntax."

[10] We do not prove formally that these concepts cannot be defined in the restricted language. Most of these are based on expressive constructs explicitly intended to be inexpressible by the language designers.

Disjunction: The restricted KL-ONE-style languages do not permit expression of concepts with disjunctive definitions. For example, these languages cannot be used to define legally employable persons. In the US, legally employable persons are persons who are of employable age and are either legal immigrants or citizens.

$$\begin{aligned} &\forall x \text{ legally-employable-person}(x) \\ &\quad \equiv \quad \text{employable-age}(x) \wedge [\text{legal-immigrant}(x) \vee \text{citizen}(x)] \end{aligned} \tag{1}$$

Disjunctively defined concepts are very common. Some other examples include major-party candidates, which in the US are candidates of either the Democratic or Republican parties; members of Congress, defined to be either representatives or senators; US citizens, defined to be people who either are born in the US or are naturalized; and nonplanar graphs, which for some purposes may be defined to be graphs containing either of the graphs $K(3,3)$ or K_5.

Negation and relative complements: Large numbers of concepts are defined as negations of other concepts.[11] For example, unemployed persons are persons who are not employed.

$$\forall x \text{ unemployed-person}(x) \equiv \text{person}(x) \wedge \neg\text{employed}(x) \tag{2}$$

An important special case consists of concepts defined as the residue or relative complement of a finite set of alternatives. For example, hypertension is divided into two categories: essential hypertension and secondary hypertension. Secondary hypertension is hypertension which is due to any of the several known causes (renal, endocrine, aortic, enzymatic, neurologic, drug-induced, hypercalcemic). Essential hypertension is the residual class of hypertension which is not secondary hypertension, that is, which is not due to one of the known causes.

$$\begin{aligned} &\forall x \text{ essential-hypertension}(x) \\ &\quad \equiv \quad \text{hypertension}(x) \wedge \neg\text{secondary-hypertension}(x) \end{aligned} \tag{3}$$

$$\begin{aligned} &\forall x \text{ secondary-hypertension}(x) \\ &\quad \equiv \quad \text{hypertension}(x) \wedge [\text{renal-hypertension}(x) \vee \ldots \\ &\qquad\quad \vee \text{hypercalcemic-hypertension}(x)] \end{aligned} \tag{4}$$

Other examples include independent voters, which are voters who do not belong to any party; independent candidates, illegal employees, unhappy persons, unsolved problems, leftovers, untreatable diseases, ineligible persons, aliens (non-citizens within a national jurisdiction), idiopathic diseases (diseases of unknown cause), splinter or fringe parties, spontaneous (unplanned) acts, eligibility of a couple to marry (no close consanguinity).

Conditionals: Concepts defined using conditionals can be divided into three main classes: ones involving only material implication, concepts involving time implicitly, and concepts

[11] As noted in section 1.4, the disjointness and cover constructs do not allow us to express negation.

explicitly depending on time. We give examples of each of these.[12] In addition, there are numerous notions of "deviant" conditionals: hypotheticals, counterfactuals, relevance logic implication, constructive implication, etc. We will not provide examples for these here.

Material implication: A problem is NP-complete just in case every problem in NP can be reduced to it in polynomial time.

$$\begin{aligned} &\forall p \text{ NP-complete}(p) \\ &\quad \equiv \quad \text{NP}(p) \wedge \forall p' \text{ NP}(p') \supset \text{P-reducible}(p', p) \end{aligned} \tag{5}$$

Implicitly temporal implications: An animal is dangerous (to humans) if it attacks them when nearby.

$$\begin{aligned} &\forall x \text{ dangerous-animal}(x) \\ &\quad \equiv \quad \text{animal}(x) \wedge \forall h \text{ human}(h) \wedge \text{near}(x, h) \supset \text{attacks}(x, h) \end{aligned} \tag{6}$$

Explicitly temporal implications: An "urgent message", according to Brachman and Schmolze [8], is a message whose reply-by date is less than one hour later than its received-by date. This is an example of an explicitly temporal conditional, one in which the conditions themselves refer to specific instants of time.

Other examples of concepts defined using conditionals include all sorts of "secondary" concepts, such as an actor's understudies, assistants, overflow tanks, safety valves, beneficiaries, resources or services such as inventories, stocks, supplies, service personnel, repairmen, many more special relationships, such as partnerships and mutual aid societies; predictive concepts, such as susceptible people (who become infected if exposed, or who buy the product if they see the ad) and effective therapies (which cure their patients if applied); and temporal concepts such as annual reports and clairvoyance.

Equivalence: Many concepts are defined by the coincidence or equivalence of two other concepts. For example, a dinner wine is proper just in case it is red whenever the main dish is red meat or is tomato based.

$$\begin{aligned} &\forall d, w \text{ proper-dinner-wine}(d, w) \\ &\quad \equiv \quad \text{dinner}(d) \wedge \text{wine}(w) \wedge \\ &\qquad\qquad \text{red-wine}(w) \equiv [\text{red-meat}(d) \vee \text{tomato-based}(d)] \wedge \\ &\qquad\qquad \text{white-wine}(w) \equiv \neg[\text{red-meat}(d) \vee \text{tomato-based}(d)] \end{aligned} \tag{7}$$

Other examples include the concepts of complete inference procedures, comparable worth, and monogamous marriages or faithful spouses.

Particularization: Particularization [1, p. 145] means specializing a concept by fixing a role's filler rather than by restricting a role's type. For example, redheads are people whose hair color is "red."

$$\forall x \text{ redhead}(x) \equiv \text{person}(x) \wedge \text{hair-color}(x) = \text{"red"} \tag{8}$$

[12] The restricted KL-ONE-style languages provide means for expressing some conditional definitions through the `CRestrict` construct. But this can only be used to express some roles, not concepts.

Other examples include the concepts of MIT alumnus and 10-foot pole.[13] See Brachman [1, p. 145] for further examples.

Recursive definitions, transitive relations, and ordered sets: Many concepts are defined recursively. For example, a person is at risk for infection if he is in contact with someone who is either infected or at risk for infection.

$$\begin{aligned}&\forall x \text{ at-risk-for-infection}(x)\\ &\quad\equiv \quad \exists y \text{ in-contact}(x,y)\\ &\qquad\quad \wedge [\text{infected}(y) \vee \text{ at-risk-for-infection}(y)]\end{aligned} \tag{9}$$

Other examples of concepts with recursive definitions include data-structures such as lists, sequences, and trees. See, for example, Schmolze's [25, p. 3] definition of binary trees.

If we look to defining relations as well, many transitive relations such as connectedness, reachability, liability, and allegiance are most naturally defined using recursive definitions.[14] But perhaps the most common case is that of relations which order sets. Many domains of reasoning employ ordering relations, such as part-whole, before-after, and cause-effect relations, which are specified as the transitive closure of primitive relations. Other examples include authority, ownership, control, and ancestry.

Functions over ordered sets: Many concepts involve functions or relations defined over ordered sets. For example, a person is eldest in a group if there is no older person in the group.

$$\begin{aligned}&\forall x, g \text{ eldest}(x,g)\\ &\quad\equiv \quad x \in g \wedge \forall y\ y \in g \wedge x \neq y \supset \text{older}(y,x)\end{aligned} \tag{10}$$

Other examples include concepts involving functions which pick out elements that are maximal, minimal, best, worst, biggest, smallest, winners, losers, average, mediocre, most, or least. Some concepts recast one or more order as another, such as the rational-choice ordering of alternatives by expected utility.

Sequences constitute another class of functions over ordered sets, specifically, those over the ordered sets of integers or natural numbers. The restricted languages do not provide any way of defining sequences, nor of defining standard functions over sequences, such as limits and first, second, last, and middle elements.

Mappings between ordered sets: Many concepts concern the relationships that hold among different ordered sets. For example, a tax schedule is progressive if the tax rate is an increasing function of income. Assuming that schedules are numerical functions of income levels, we may define this concept as follows.

$$\begin{aligned}&\forall s \text{ progressive-tax-schedule}(s)\\ &\quad\equiv \quad \text{tax-schedule}(s) \wedge \forall x, y\ [x \leq y] \supset [s(x) \leq s(y)]\end{aligned} \tag{11}$$

[13] Particularization is allowed in KANDOR but not in NIKL, except by introducing a primitive individual concept such as "10 feet."

[14] NIKL's role chains can express transitive information, but role chains and RVM's cannot be used to *define* transitive relations because the attempt to do so makes the classifier go into an endless recursion.

As the definition indicates, progressive tax schedules are simply types of monotone increasing functions. Monotone increasing and decreasing functions underlie large numbers of common concepts in addition to progressive taxation, such as regressive taxation, seniority privileges, proportional representation, vacation time benefits, progressive speeding fines, causal influences, and proportionalities.

Functions of sets and functions: Functions over ordered sets, examined previously, are just one class of functions over sets. Another large class of functions over sets are numerical measures. Roles whose fillers must be single numbers represent functions over the concept's instances. Common applications go beyond stating the mere existence of these functions by employing concepts that are essentially measures defined over sets of instances. For example, the weight of a shipment is the sum of the weights of all of its items.

$$\forall x \text{ shipment}(x) \supset \text{weight}(x) = \sum_{i \in \text{items}(x)} \text{weight}(i) \tag{12}$$

In addition to measure-like concepts, such as net worth and test scores, concepts such as directions or rates of change (acceleration, spurts, slowdowns, speedups, inflation, deflation, etc.) are defined as more general functions over numeric roles.

Finally, some concepts are defined as functions over sets which need not be ordered or numerical. One broad class is that of sets of representatives drawn from sets of sets. For example, a balanced jury is one which contains jurors from each of the main social groups of the citizenry (whatever those happen to be at the time), and a broad-spectrum political commission is one which contains members from each of the interested political-action groups or parties.

Binary functions, n-ary relations: KL-ONE-style languages supply directly only two sorts of logical entities: unary predicates (concepts) and binary predicates (roles). There is no direct provision for expressing binary functions, ternary predicates, or any functions or predicates of higher arity. For example, the arithmetic addition function may be defined in terms of the successor function s and predecessor function s^{-1} as

$$+ = \lambda x, y \text{ IF } y = 0 \text{ THEN } x \text{ ELSE } s(x) + s^{-1}(y)$$

but this definition is not expressible in the restricted languages. Other undefinable functions and relations include concepts of betweenness, conduits, supports, escape routes, connected routes, and most assemblies of parts and wholes. Indeed, defining the classic AI illustration of an arch of toy blocks involves defining interrelations among its structural components (the vertical and horizontal clearances, for instance—see [44, p. 52, Fig. 5]) that cannot be expressed in the restricted language. This example, in fact, serves as a central source of examples for Brachman's argument (see [2], [3, p. 35ff]) stressing the need for specifications of the structural relations between role fillers.

One can use unary and binary predicates to name the aspects of higher arity functions and relations, with each place represented by a role, e.g., the addend, augend, and sum roles of an addition. But simply naming these places, and perhaps restricting the types of their fillers individually is the most that the restricted languages can do. This does not and cannot by itself actually define the relation that is supposed to hold among these places. For example, simply defining the roles of the addition relation does not define the mathematical function of addition. More generally, some higher arity relations used in definitions are usually taken to be primitive, and these cannot be used with the restriction to unary and binary predicates.

2.2 Assessing the impact on applications

These examples substantiate the fact that the language restrictions are indeed significant. Moreover, such restrictions incur a large cost to developers of applications in working around restrictions, or inventing mechanisms external to the language to meet the needs of their applications.

The designers of restricted language representation systems expected that the concepts expressible in these languages would include those necessary for practical applications. The several successful NIKL applications to limited tasks of natural-language understanding, automatic programming and planning encouraged this expectation. However, at least two separate research efforts using NIKL, those of Haimowitz, Patil, and Szolovits [13] and Smoliar and Swartout [39], have found these restrictions to be very serious impediments to the development of useful knowledge bases. As the broad scope of the examples indicates, the difficulties these efforts experienced are merely the tip of the iceberg. It is not possible to work around these language restrictions in practice, because these restrictions were specifically chosen to make certain conceptual relations inexpressible.

Language restrictions mean that the user must resort to ad hoc techniques (such as using inherited data lists as property lists) in order to represent information which is not of the form directly supported by the language. Unfortunately, different users may invent different ad hoc mechanisms for different applications, even when the target knowledge and inferences are identical. An example of this difficulty is provided by the NIKL reimplementations of XPLAIN and ABEL. XPLAIN [40] generated causal explanations of digitalis therapies, and ABEL [33] generated causal explanations of acid-base electrolyte disorders. Both systems were originally implemented using XLMS [14]. Each used the same notion of causality in its knowledge base, and the original knowledge bases were sufficiently compatible to allow the two programs to share some of the same utilities (among others, the natural language generator). The NIKL reimplementations lacked this compatibility, however, since many of the representational requirements of these systems were not available in NIKL, and the different translators came up with different ad hoc techniques external to the representation system to embody this knowledge. Any language of limited expressiveness will cause similar problems, since while different users are likely to formulate expressible knowledge in compatible ways, nothing forces them to encode inexpressible knowledge in compatible ways. Even when they cannot be exploited as efficiently via classification, expressive languages provide standards or conventions for representing knowledge independent of particular inference mechanisms and applications.

Even if one is willing to live with incompatible representations that cannot be shared among applications, inventing new external mechanisms is a lot of work that must be repeated for each application. For example, Smoliar and Swartout found this effort so burdensome that they developed a more expressive language [39] to use instead of NIKL.

Such extended languages, of course, mean that the classifier must be either incomplete or excessively slow in the worst case.

Levesque and Brachman's arguments for excluding certain constructs from the concept definition language were based on the need to ensure the (real-time, worst-case) responsiveness of the knowledge base for general purpose use. As we have seen, these exclusions render these systems unsuitable for general purpose use. The moral is that the tradeoff between expressiveness and complexity means that there is no general purpose language. Any language expressive enough to state the knowledge needed in each application will cause the classifier to fail to perform adequately in some applications, and any language restricted enough to permit tractable classification in all applications will fail to permit expression of concepts necessary to some applications.

3 Inexpressibility, primitive concepts, and classification

The desire for efficient classification motivated language restrictions. However, the interaction of these restrictions with primitive concepts reduces the utility of classification.

Almost every application domain is conceptualized using many *natural kind* concepts, which cannot be defined in terms of necessary and sufficient conditions. Common and exotic natural kinds form the backbone of taxonomies. Most of the everyday nouns in use in natural language (e.g., mammal, dog, table, shock, thud, etc.) refer to natural kinds, and these support commonsense reasoning. In addition, many of the terms used in specialized fields (e.g., mass, energy, chaos, symbiosis, market, workstation, etc.) are also natural kinds, albeit only within a restricted community of experts. Because they cannot be precisely defined, natural kind concepts must be entered as primitives. Thus most knowledge bases will contain many primitives.

Primitives, however, reduce the utility of classification because they are not classifiable. This effect is magnified because subconcepts of primitives cannot be classified unless they explicitly contain all their primitive superiors. For example, the concept "cat" may be a primitive specialization of other concepts which describe the properties of cats (four-legged, furry, clawed, etc.), and the concept "black cat" may be defined as cats which are also black. But to classify a concept description as black cat one must first determine that it is a cat, even if the description contains every property of cats plus blackness. Lacking the explicit assertion that it is a cat, the new description cannot be classified as the black cat concept.

Language restrictions make it necessary to treat many definable concepts as primitive as well. Any inexpressible concept must be introduced as a primitive if it is to be used at all. Yet these "fake" primitives are just as unclassifiable as natural kinds, even though their use is an artifact of the language restrictions.

It appears that for many applications, a large majority of concepts must be entered as primitives in KL-ONE-style languages. Some support for this assessment comes from noticing that the great majority of concepts are primitive in many of the sample networks published by Brachman and Schmolze [8]. These diagrams, however, may have been simplified for publication by leaving out some expressible concepts. Stronger evidence comes from the previously mentioned project of rewriting in NIKL the ABEL system for acid-base electrolyte diagnosis (see [13]). This effort revealed that fewer than a third of the concepts in this knowledge base were classifiable, as most important concepts referred either to domain-specific natural kinds or to inexpressible concepts. A similar experience is related by Smoliar and Swartout [39]. Judging by these cases, KL-ONE-style languages significantly restrict their languages in order to speed up an operation applicable to only a small fraction of concepts.

4 Classification and assertional information

The restriction of classification to purely terminological information reduces the usefulness of classification because the proper categorization of some definable concepts depends on contingent information. For example, the concept of treatable disease is straightforwardly definable: treatable diseases are diseases for which there exists a treatment, or (`CMeet` DISEASE (`CMin` TREATMENT 1)). However, the information about treatments that distinguishes treatable from untreatable diseases is usually viewed as assertional, and so would not appear in a purely terminological database. This means that the classifier cannot recognize classes of diseases for which there are treatments available (such as bacterial infections) as subconcepts of treatable disease. For instance, acute glomerulonephritis (AGN) is a type of disease. But since the statement that AGN is treatable is true but not definitional for AGN (and so appears in the assertional knowledge base), purely terminological classifiers will not classify AGN as a type of treatable disease. To achieve the desired classification, we must add TREATABLE-DISEASE to AGN's definition, in effect asserting that AGN is a type of treatable disease.

In fact, there are large numbers of important concepts which are similarly defined over contingent information. These include the concepts of solvable problems, childless persons, feasible schedules, winnable positions, illegal acts, constitutional laws, industrial nations, and democratic countries. In some of these cases, the concept's extension represents historical information that cannot be changed by future events. For example, the discovery of quinine and penicillin changed malaria and syphilis respectively from untreatable to treatable diseases. In other cases, the extension's changes are not cumulative, but the extension changes slowly enough to make extensional classification worthwhile. For example, the sets of developed, developing, and less developed countries may lose members as well as gain them, but such changes occur infrequently enough that everyday references to these categories may make good use of the extensional classifications.

In each of these cases, extensionally defined concepts must be treated much like primitives, since proper classification of both is determined by contingent facts. Recall that primitive concepts cannot be properly classified as subconcepts without the user specifically indicating that they are specializations of other concepts. This is essentially assertional information, but of a kind tolerated in the terminological database. Extensionally defined concepts differ only in that the assertional information which indicates their proper classification is usually not tolerated in the terminological database. One could instead allow more assertional information to be reflected in the terminological database, but this destroys the separation of terminological and assertional knowledge, and can make definitions unnecessarily complex.

5 Omitting versus limiting constructs

Even if the argument for ruling out bad worst-case behaviors is sound, the restrictions placed on the concept-definition language seem unnecessarily harsh. Specifically, it need not be necessary to forbid the use of certain constructs. Merely limiting their use might also rule out the unacceptable worst-case behaviors if the common uses of the problematic constructs are very limited and efficiently classifiable.

To see this, compare the situation in knowledge representation with that of parsing natural languages. Natural languages typically permit center embedding of clauses. When this observation is taken to hold uniformly, natural languages must be at least context-free languages, and so require time at least cubic (roughly) in sentence length to parse. On the other hand, we seem to be able to parse and comprehend sentences in these languages in real time. Applied to the design of natural languages, the restricted language thesis would indicate forbidding center embedding (or perhaps concatenation). Of course, there is a middle way. While we can understand center embedded sentences, we cannot comprehend sentences with more than three or four levels of embedding. Thus grammars of spoken languages that permit center embedding, but only to three or four levels, maintain the regular (real time) character of spoken language yet allow the common usages of the potentially troublesome expressive mechanism. The fact that comprehension must occur in real time need not be taken as an argument against the use of a context-free grammar to describe the basic structure of the language.

This suggests that efficient knowledge representation services need not completely forbid dangerous constructs, but may only need to limit their usage. For example, permitting bounded numbers of bounded-size disjunctions may preserve tractability. It may even be sufficient to instill a sense of caution in the user when using the constructs.

6 Classification's completeness

The demand for completeness of classification is fundamentally a demand that the terminological component know what its terms mean, that it be able to recognize equivalent definitions in spite of different phrasings or syntactic presentations.[15] As Brachman and Levesque [7, p. 192] put it, "competence in the terminological component depends on closure under subsumption (i.e., that the system "knows" when one term conceptually contains another)." "Knowing," in this sense, amounts to possessing a full understanding of all the consequences of definitions.

Levesque and Brachman [19, p. 79] motivate this demand in terms of the goal of *shared understanding*, which means that the knowledge base understands the user's meanings and the user understands the knowledge base's meanings. But since efficient classifiers are unable to fully appreciate all the implications of some definitions, the knowledge base can understand the user's meanings only if the terminological language is restricted so that complete classification is possible over the restricted language. Drawing on Smith [38], Levesque and Brachman state that

[15] It is worth noting that the incompleteness of NIKL's classifier has not seemed to matter in practical applications. Empirically, it is complete enough.

> "[t]he symbolic structures within a knowledge-based system must play a *causal role* in the behaviour of that system, as opposed to, say, comments in a programming language. Moreover, the influence they have on the behaviour of the system should agree with our understanding of them as propositions representing knowledge. Not that the system has to be aware in any mysterious way of the interpretation of its structures and their connection to the world; but for us to call it knowledge-based, *we* have to be able to understand its behaviour as if it believed these propositions, just as we understand the behaviour of a numerical program as if it appreciated the connection between bit patterns and abstract numerical quantities" (authors' emphases).

Thus the requirement of shared understanding rules out the use of definitions that cannot be classified at all or rapidly enough, for if they are permitted, the classifier must be incomplete, and if they are ignored, they are like "comments," representations not understood by the system itself.

We believe that the demand that definitions be fully understood by the classifier (in the sense that all their consequences are known) is much too strong because it requires complete understanding as a prerequisite to use. While it may be desirable to demand that knowledge bases and their users share understandings of meanings, understanding of meanings may be shared even if these understandings are incomplete.[16] A more reasonable demand is for shared knowledge. As the terms are employed in ordinary discourse, there is a great gulf between "knowing" and "understanding," with knowing a much weaker notion than understanding. For example, teachers commonly observe that students can know facts or formulas, but not understand them. That is, students may remember the information they have been given, but many do not fully understand the implications of what they have been told. Indeed, in some cases students may be taught how to perform various formulaic operations mechanically prior to being taught what these formulas and operations are for. Everybody, of course, knows that $E = mc^2$, but how many understand it? Against this ordinary usage, the goal of terminological competence confounds knowing and understanding.[17]

As Minsky [23, p. 128] observes, "*thinking begins first with suggestive but defective plans and images that are slowly (if ever) refined and replaced by better ones.*" This is especially true of almost all learning and knowledge engineering tasks. But if knowledge representation services are to support imperfect and developmental reasoning and learning, there is no alternative to storing and retrieving information that is not yet completely understood. This is obvious to everyone in the context of assertional information, where no one supposes that a system should forbid adding any axiom for which it cannot divine all the entailed consequences. Why should the situation be different for definitional information?

Restricting languages to ensure classification's completeness turns the usual relation between knowledge bases and databases on its head. In the usual conception, databases can store any scrap of information, just like a desk drawer, while knowledge bases go beyond databases by adding some inferential capabilities. But restricted language knowledge bases

[16] Shared understanding certainly is no prerequisite for communication. Teachers, for example, are usually expected to understand much more than their students.

[17] This confusion is also well established in epistemic logics, which typically require that knowing something means knowing all its consequences.

cannot accept or store anything they don't understand, while databases can store information whatsoever. In this conception, restricted language knowledge bases need not be augmentations of databases at all, but something both more (in the inferences they support) and less (in the content they permit) than databases.

7 Classification's efficiency

Levesque and Brachman's argument for the restricted language thesis views representation services as subroutines to which the agent (temporarily) surrenders control [19, p. 81]. Because the reasoner using the representation service must be able to act when necessary, they conclude that the terminological language must exclude any construct that can lead to large polynomial or exponential computation times during calls to these subroutines. But subroutines are very special forms of services. In everyday life, services are tools the agent may discard in mid-use if they do not perform satisfactorily. This is true even of computers, which usually come equipped with an "abort" key so that the user may exit computations prematurely. Properly viewed, Brachman and Levesque's concern for freedom of action does not argue against bad worst-case behavior as much as argue for treating representational services as separate processes, with query transactions interruptible as the needs of the reasoner demand. This makes utilities slaves to reasoners, rather than making reasoners temporary slaves to utilities.

But even if representation services are subroutines, judging failures by their worst-case time complexity misconceives both the nature of classification's efficiency and its role in introducing language restrictions. We examine each of these in turn. We begin by discussing some difficulties inherent in using worst-case time costs as a measure of classification's efficiency. We then question whether efficiency is the proper standard for judging inferences in the first place.

7.1 Measuring efficiency

The standard of worst-case time efficiency is motivated by the desire to provide a general purpose representation service which is of use in all applications, and hence quick enough even for the most critical. The first problem with measuring classification's efficiency in this way is that this underlying goal is not well defined, since for any specific representation system one can always find some task which requires faster response than it can guarantee. The demand that classification always operate swiftly seems even less justified when we consider that the general query posed to a knowledge representation service is not posed directly to the terminological component but to the representation system as a whole. This means that answers to general queries may depend on the assertional component as well. Since inference over the assertional knowledge base need not even be decidable, failures of queries to receive prompt answers seems inevitable.

The second problem with identifying efficiency with worst-case time costs is that the worst case is not the general case. The restricted language thesis judges failures by their worst consequences in the most critical applications, independent of how rare or common these failures are. Such judgments may be a reasonable design specification in some exceptionally critical applications in which failure to produce an answer quickly has horrible consequences, but such catastrophic applications are rare among typical artificial intelligence applications. Instead of being the rule, critical applications call for their own special design criteria which cannot be expected to form the basis for the everyday exercise of intelligence. How many jobs do we give humans in which we promise to fire or kill the worker if he makes any mistake or fails to meet any deadline by any amount in his performance of routine tasks? But if we are willing to tolerate the occasional late or missing answer, there is no reason in principle that we should forbid services whose worst case is not what we prefer.

While Levesque and Brachman do not explicitly consider all ways of taking into account the probabilities of encountering worst-case behaviors, they explicitly reject the suggestion that efficiency be measured by the average time needed for classification [19, p. 81]. They reject this possibility in part because little is known about reasonable distributions over queries that would, for example, allow average times to be calculated for different algorithms, or even how to calculate such averages if we knew the distributions. (But see Goldberg [12], who demonstrates that the exponential worst case satisfiability problems are sparse, and that average complexity is polynomial.) Contrary to Levesque's and Brachman's claim, however, this theoretical ignorance need not be a serious impediment to reliance on average response times since we may summarize practical experience in estimates of average running times, periodically revising these estimates to provide an acceptable level of accuracy.

Indeed, the prevalence of unclassifiable primitive concepts noted earlier reduces the likelihood of encountering worst-case classification tasks. For example, NIKL's classifier requires exponential time in the worst case, which occurs when the concept lattice consists entirely of defined concepts. In practice, however, it responds quickly since the wide distribution of primitive concepts means that its search span is severely restricted. This means that worst-case complexity is not a good measure of efficiency when large numbers of primitives are present, for if classifiable concepts are rare, it does not matter whether they are treated very efficiently. Thus the prevalence of primitive concepts undermines the argument for restricting the concept-definition language to ensure efficient classification.

The third problem with worst-case time is that this measure of efficiency ignores the other costs (e.g., space consumption) incurred in reasoning. This means that the optimum system performance may not be achieved purely by optimizing the time costs of classification. A case in point is that language restrictions impose intellectual costs on developers of applications, who must expend effort to formulate their applications in a language designed to make this difficult. During the developmental process, the effort expended by the user or reasoner in incrementally formulating, developing, testing, and refining the desired knowledge and performance is more significant than the instantaneous time costs of classification. In this situation developmental efficiency is the primary concern, and it is disadvantageous to trade limited expressiveness and the added developmental effort it incurs for a guarantee of runtime efficiency.

Finally, worst-case time may mismeasure the system's efficiency because the overall costs (time or otherwise) of a system depend on more than just the costs of classification. We do not doubt that classification has an important role to play in recognition and reasoning, but general reasoning systems must provide support for a broad spectrum of services in addition to taxonomic classification, such as general logical reasoning, probabilistic reasoning, default reasoning, hypothetical reasoning, and revision of beliefs and goals. This multiplicity of functions means that the fundamental tradeoff of knowledge representation is not just that between expressiveness and efficiency of classification, but the broader tradeoff between the overall costs of different means to carry out reasoning and representation functions in different computational or cognitive architectures. We do not yet know enough to be able to assess the relative merits of different representation and reasoning facilities, but it is quite possible that designing a system to optimize classification will make these other operations unduly expensive and thereby make the overall architecture distinctly non-optimal. Indeed, we argue in section 8.3 that default reasoning is a good example of such a tradeoff.

7.2 Computational efficiency versus decision-theoretic utility

The preceding problems with using worst-case time costs to select linguistic restrictions are important, but they are all symptoms of a more profound problem. No matter how one measures the costs of classification, costs alone say nothing about the value or benefits received from classification. If one is to make design choices as consequential as restricting the terminological language, one must take the benefits of classification into account as well as its costs. This means judging classification actions by the standard of decision-theoretic rationality.

Formally speaking, one must judge the designs and actions of representation systems according to their decision-theoretic utility rather than their computational costs. The utility of some action is usually not identical to the cost of the action, but to some function of the cost and the other consequences of the action. For the case of a representation service, the value of an answer usually depends on numerous variables: on what the question was, in what circumstances it was asked, on when the answer is received, and on how reliable the answer is. In addition, the value of classification or other inferences also depends on the present value of their future utility, as implied in our earlier discussion of developmental costs. The answer's value may vary discontinuously with changes in some of these variables. For response time, however, the usual case is that its value declines smoothly (though perhaps rapidly) with time, rather than undergoing the sudden transition from acceptable to absolutely unacceptable that the worst-case time measure suggests.

Rationality, in the sense of decision theory, means taking actions of maximal expected utility. The expected utility of an action is the utility of its consequences in each situation averaged according to the probabilities of the consequences occurring in those situations. Just as utility is usually not the same as cost, expected utility is not usually identical to average cost, even when utility is a function of cost alone. Expected utility necessarily averages over utilities, not over the variables on which utilities depend. For example, bicycles designed to fit the average size rider perfectly poorly serve a bimodal population of tall adults and short children. In the same way, expected computational utility need not be a function of average running time and average space.

Judging classification and other inferential operations according to their expected utility instead of by their worst-case time cost recognizes that all actions may fail with some probability, that failures have different consequences in different circumstances, and that probabilities and utilities of different consequences vary independently. By permitting utility to vary with the situation, this standard encompasses ordinary actions in which failures are tolerable as well as critical actions in which any failure is catastrophic. Using the notion of utility, we may rephrase the restricted language and restricted classification theses as stating that the expected utility of systems designed with these restrictions dominates the expected utility of other designs. But no one has ever made detailed arguments for these modified theses, since the focus has been on computational costs instead of on computational or cognitive utility.

8 Some alternative approaches

The arguments of this paper suggest that in order to be general purpose the system should provide a language expressive enough to permit definition of all important concepts of most domains of knowledge. The emphasis should be on expanding expressive power rather than minimizing computation time. Attempting to avoid excessive computation through linguistic restrictions is analogous to attempting to avoid buggy programs by designing programming languages which guarantee correct execution. The latter certainly seems impossible; what works better is for programmers to know when they are using a dangerous construct.

Our arguments also suggest explicit provision for incomplete deductive classification, classification over specific categories (perhaps user-specified) of relevant assertional knowledge, nondeductive or "approximate" forms of classification of concepts involving defaults, and rational management of inference tools.

8.1 Tolerating incomplete classification

We believe that general purpose systems should tolerate incomplete classification since they already tolerate enormous degrees of incompleteness in general knowledge and reasoning. Two ways of doing this involve restricting the notion of completeness, either by seeking completeness with respect to weaker logics (see, for example [31]), or by seeking completeness with respect to ordinary deduction over sublanguages (see, for example [10, 11]). Virtually all the work on restricted language classification can be immediately reformulated as a contribution toward this end.

Another useful approach is to permit the user to directly assert subsumption relationships in cases too hard for the classifier to determine, as is already done (in a sense) when the user defines primitive specializations of concepts. In some cases, whole classes of subsumption relationships may be asserted at once. This commonly occurs when nonlogical relations between concepts entail subsumption relationships between other concepts. For example, nephrons are parts of kidneys, and nephrotic diseases are types of kidney diseases. The former fact does not logically entail the latter. Instead, these two facts represent a single instance of a general principle about diseases: that diseases of parts of organs are also diseases of the organs themselves, or formally,

$$\forall x, y, d \ \text{organ}(x) \wedge \text{organ}(y) \wedge \text{part-of}(x, y) \\ \supset [\text{disease-of}(d, x) \supset \text{disease-of}(d, y)].$$

It would be very valuable to be able to tell the classifier this principle so that it could explicitly recognize these new subsumption relationships.[18] Indeed, there are many other nonlogical relationships (especially transitive relations like temporal precedence, causality, parts and subparts, and hierarchies) which indicate important classes of subsumption relationships in this way.

Other cases involve multiple definitions, which are useful when different intellectual tasks are easier using one definition rather than another.[19] For example, to teach people the concept of nonplanar graph, the proper definition is the straightforward one: a nonplanar graph is a graph which is not homeomorphic to any subset of the plane. But to recognize nonplanar graphs, a much better definition is one based on Kuratowski's theorem: a nonplanar graph is one which contains a subgraph homeomorphic to one of the graphs $K(3,3)$ or K_5. Such nontrivial alternative definitions are nontrivial because proving their equivalence is not easy. When these equivalences are known, there is no reason to burden the classifier with the chore of rediscovering them.

8.2 Classification over relevant knowledge

It is important to broaden the scope of classification to include relevant assertional knowledge because the most useful classifications of some concepts do not always respect the division of knowledge into terminological and assertional components. When most reasoning about definable concepts makes use of their contingent extensions, classification should aid reasoning by using those extensions in organizing the subsumption lattice. Classification need not be burdened with the task of inventing the concepts or their extensions: that is up to the user, or to learning and problem solving procedures, which determine which concepts are useful in general and which are the appropriate abstractions for specific reasoning purposes. The classifier need only take into account specific classes of assertions that are explicitly indicated by the external reasoner or user. But when learning and problem solving procedures determine sets of interesting concepts and relevant sorts of contingent facts, the classifier should classify new concepts with respect to them. Classification over relevant knowledge is well within the scope of existing architectures, but does raise interesting problems about how to modify classifications based on changing knowledge or assumptions.

8.3 Nondeductive recognition and defaults

Another facility we believe too valuable to omit is provision of nondeductive forms of "approximate" classification or *categorization*. (We will use this broader term to avoid the logical connotation of classification.) For example, ordinarily something can be recognized as a duck if it walks like a duck, looks like a duck, and quacks like a duck. In this case, the most useful categorization is one that provides the best guess about the category of the concept being described, even though this guess may be approximate rather than exact, and may ignore details to get the main points right.

For much the same reasons, classification is much more valuable when concept definitions are allowed to include default properties. Default properties are widely considered to be a

[18] CAKE [35], for example, offers this ability. In some cases, specifying the class of subsumption relationships may require the ability to quantify over concept definitions.

[19] Synonyms and abbreviations constitute trivial but useful forms of multiple definitions.

central element of taxonomic inference, mainly because there seem to be very few strictly necessary conditions in most natural kinds (cf. [5, 16]). Instead of being definable by sets of necessary and sufficient conditions, natural kinds appear to be conceptually convenient bundles of default properties which describe the prototype to which their instances bear family resemblances.

Though the prevalence of natural kinds and the importance of recognizing their subkinds and instances suggests that defaults in definitions are too important to leave out, default properties are explicitly excluded from KL-ONE-style languages. In part, this is due to the requirement that classification perform only sound, correct inferences. In addition, Brachman and Schmolze [8, p. 189] argue that defaults and exceptions should not be part of the terminological database because they make classification too hard. This argument would have more force if the tractable part of classification provided more useful answers than the tractable part of classification or categorization over concepts with defaults. But as we saw above, the fragment of deductive classification provided by KL-ONE-style languages is very limited. Indeed, recent results of Patel-Schneider [32] and Nebel [28] mean that even these languages are too rich, according to the restricted language thesis. It seems likely that incorporating some fragment of default reasoning could only increase the utility of classification.

8.4 Rational management of inference tools

KL-ONE-style representation systems are intended to be general purpose, domain independent tools for use in all sorts of reasoning tasks. We suggest that to design a general purpose, domain independent representation system one must make it gain its purpose from its user (by which we here mean either the external reasoning system or a human user), not from its designer. A designer can only pick a point on the expressivity-efficiency tradeoff that is appropriate for the purposes of some applications but not for the purposes of others. What is needed instead is for the representation service to cooperate with the user in achieving the user's aims. The user's aims include keeping track of various information, and finding out about this information and its consequences. To help the user do these things, the system needs to know what its user knows, and also needs to know what its user wants when the user asks questions of it. One may view the relation between the representation service and its user as analogous to the relation between a librarian and a researcher. Librarians are useful to researchers because they have (or try to obtain) some knowledge of what the researcher wants and use that knowledge in deciding how best to assist the researcher. Asked for information on some topic, the librarian might provide the requested information immediately. But the librarian might instead immediately respond with "I will not be able to help you with that," or "I can help you, but it will take some time, if you can wait." Similarly, a representation service should be able to tailor its answers according to the importance and immediacy attached to them by the user and according to the resources at its disposal for calculating the answers.

Viewed in this way, the function of a general purpose representation service is to rationally manage the application of a set of inference tools to a store of knowledge. The knowledge base manager is a specialized agent (limited in the range of its knowledge and actions) that rationally allocates its efforts in answering questions. The key point is that the knowledge base does not have preferences of its own, but instead obtains or adopts its preferences about actions from the user. For this to be possible, the user's commands or queries must indicate the purpose of the operation as well as the knowledge involved. Of course, the purposes or goals need not be passed as explicit parameters, but might be stored in the same knowledge base as shared goals. Obtaining expectations about tool performance and distribution of queries is more difficult. This problem might be amenable to machine learning techniques (see, for example, [24]).

9 Conclusion

Levesque and Brachman argue that in order to be useful for the most critical applications, general purpose knowledge representation systems should restrict their languages by omitting constructs which require non-polynomial (or otherwise unacceptably long) worst-case response times for correct classification of concepts. They also restrict the classifier to operate over the purely definitional information in the terminological knowledge base. We have argued in this paper that both of these theses are flawed, and that in practice the terminological facilities of systems designed along these lines are so expressively impoverished that the very purpose set out for general purpose representational utilities is defeated. Specifically, we argued that (1) restricting languages by omitting constructs which require non-polynomial (or otherwise unacceptably long) worst-case response times for correct classification of concepts destroys the generality of the language and system; (2) definitions inexpressible due to language restrictions must be entered as primitive concepts, which are unclassifiable and which reduce the utility of classification; (3) restricting classification to purely terminological information significantly reduces its utility in practical applications; (4) language restrictions which omit constructs entirely make more drastic limitations than are necessary to ensure efficient worst-case response time; (5) language restrictions aimed at ensuring the completeness of efficient classification prevents storage or retrieval of available information prior to full inferential understanding of this information; (6) requiring that general purpose systems be quick enough for the most critical applications misconceives the notion of runtime efficiency and overlooks the fact that optimizing overall system performance depends on the complete mix of operations performed, not just classification. Given Levesque and Brachman's assumptions, there are no general purpose representation systems. The tradeoff between expressiveness and complexity means that any language expressive enough to state the knowledge needed in each application will cause the classifier to fail to perform adequately in some applications, and any language restricted enough to permit tractable classification in all applications will fail to permit expression of concepts necessary to some applications.

We do, however, believe that general purpose representation systems are possible. The key to constructing them is to recognize that worst case time complexity (or for that matter, logical soundness and completeness) is not the right measure for evaluating the services provided by representation services and to abandon linguistic restrictions based solely on this measure. Instead, the representation system should be designed to offer a broad mix of services varying in cost and quality, and to ensure that the specific responses it delivers are cost-effective by taking into account the costs and benefits of these responses as perceived by the system's user. In particular, representation systems should augment deductive terminological classification with provisions for incomplete classification, terminological classification over relevant information (whether contingent or definitional), and nondeductive forms of categorization permitting "approximate" classification and "classification" of concepts involving defaults.

To summarize, the goal of KL-ONE-style languages is to be general purpose representational systems, but the restricted language and restricted classification designs developed for these languages move them away from this goal rather than toward it. These designs give classification the central role in determining the organization of representation and inference. Our arguments show that though classification is useful as a special purpose subsystem, giving it this central role destroys the generality of the overall system.

Acknowledgment

This paper is based in part on earlier research conducted by Ira Haimowitz, Ramesh Patil, and Peter Szolovits, reported in [13]. We are specially indebted to Randall Davis and Peter Patel-Schneider for extensive, helpful comments on earlier drafts. We also thank Fahiem Bacchus, Ronald Brachman, David Israel, Yeona Jang, Hector Levesque, Elisha Sacks, James Schmolze, Oksana Senyk, Howard Shrobe, William Swartout, Peter Szolovits, Michael Wellman, and Thomas Wu for valuable comments and suggestions. The opinions expressed, however, are those of the authors, not necessarily those of these commentators. This research was supported by the National Library of Medicine under National Institutes of Health Grant No. 5 R01 LM04493.

Addendum

While we were reviewing the galley sheets for this paper, comments from additional readers raised several points for clarification.

The first two points concern the nature of rational management of inference tools. Our proposal is that the responses given by knowledge representation systems should, ideally, be rational from the perspective of the user or external problem solving system. Rationality here means that responses should be appropriate to the needs of the questioner, in a sense made precise by decision theory as "maximally preferred" according to the preferences of the questioner. This sense of rationality often raises concerns about the computational feasibility of performing explicit decision-theoretic optimizations. But our proposal is not that the manager engage in explicit decision-theoretic calculations, only that the methods it employs provide answers that are rational when judged according to the purposes and preferences of the questioner. The present paper does not attempt to present any means for computing rational or approximately rational responses, other than to point out that the relevant information about preferences and purposes may already be available for queries posed by external problem solvers.

Another concern is that the decision-theoretic sense of rationality makes "maximizing preferability" an imperative overriding all other considerations. But this concern confuses overall judgments of preferability with the various individual considerations that play some role in making these overall judgments. Rational management does not presuppose or require adoption of any single criterion for evaluating the utility of representational systems. Indeed, we criticize the restricted language thesis precisely because it calls for making worst-case response time, in the most critical situations, into the primary criterion according to which one judges representation systems. Because it calls for subordinating all other measures of costs, benefits, efficiency, and utility to this one measure, it yields, as our arguments show, only representational systems that do not deserve the title "general purpose" (however useful these systems may be for some specialized tasks). Our proposal for rational management simply attempts to draw on decision theory's explicit recognition that preference (or utility) can depend on many different attributes of a system's behavior. According to decision theory, what responses are rational to the user depends only on the user's preferences about responses; it does not depend on what attributes of responses the user takes into account in adopting these preferences. This recognition of the "multi-dimensionality" of knowledge representation problems lies at the heart of Brachman and Levesque's emphasis (in [19]) on the importance of tradeoffs in choosing architectures, and we view it as one of the many points of agreement we have with their overall view of knowledge representation. We take issue with the restricted language thesis only because we view it as based on a single-issue utility measure that is inappropriate for designing general purpose knowledge representation systems.

The third point is that several versions of this paper have been circulating (under several titles) among knowledge representation researchers since the latter part of 1988 (primarily as MIT/LCS/TM-387 and 387b). Our concern in this paper is solely with the restricted language and restricted classification theses, which have continued to exert influence on the field since their first appearances in [4, 6, 7, 19]. Subsequent to these articles and the writing of our paper, however, work by a number of authors (including Brachman, Levesque, and their students) has been concerned with improving the utility of knowledge representation systems in some of the directions we urge in our discussion, for example, by making use of default information, by sacrificing logical properties for computational effectiveness, and by treating representational issues by using the full expressive power of mathematical logic.

References

[1] R. J. Brachman. What's in a concept: structural foundations for semantic networks. *International Journal of Man-Machine Studies*, 9:127–152, 1977.

[2] R. J. Brachman. A structural paradigm for representing knowledge. Technical report no. 3605, Bolt Beranek and Newman Inc., May 1978. Also Harvard University Ph.D. dissertation.

[3] R. J. Brachman. On the epistemological status of semantic networks. In N. V. Findler, editor, *Associative Networks*. Academic Press, 1979.

[4] R. J. Brachman. What IS-A is and isn't: An analysis of taxonomic links in semantic networks. *IEEE Computer*, 16(10):30–36, October 1983.

[5] R. J. Brachman. 'I lied about the trees,' or, Defaults and definitions in knowledge representation. *AI Magazine*, 6(3):80–93, 1985.

[6] R. J. Brachman, R. E. Fikes, and H. J. Levesque. Krypton: A functional approach to knowledge representation. *Computer*, 16(10):67–73, 1983.

[7] R. J. Brachman and H. J. Levesque. Competence in knowledge representation. In *Proceedings of the National Conference on Artificial Intelligence*, pages 189–192. American Association for Artificial Intelligence, 1982.

[8] R. J. Brachman and J. G. Schmolze. An overview of the KL-ONE knowledge representation system. *Cognitive Science*, 9:171–216, 1985.

[9] J. Doyle. Constructive belief and rational representation. *Computational Intelligence*, 5(1):1–11, February 1989.

[10] A. M. Frisch. Knowledge retrieval as specialized inference. Technical Report UIUCDCS-R-88-1404, Department of Computer Science, University of Illinois at Urbana-Champaign, 1988.

[11] A. M. Frisch and J. F. Allen. Knowledge retrieval as limited inference. In D. W. Loveland, editor, *Sixth Conference on Automated Deduction*, volume 138 of *Lecture Notes in Computer Science*, pages 274–291. Springer-Verlag, Berlin, 1982.

[12] A. Goldberg. Average complexity of the satisfiability problem. In W. S. Joyner, editor, *Proceedings of the Fourth Workshop on Automated Deduction*, pages 1–6, 1979.

[13] I. J. Haimowitz, R. S. Patil, and P. Szolovits. Representing medical knowledge in a terminological language is difficult. In *Symposium on Computer Applications in Medical Care*, pages 101–105, 1988.

[14] L. B. Hawkinson. XLMS: A linguistic memory system. TM 173, Massachusetts Institute of Technology, Laboratory for Computer Science, 545 Technology Square, Cambridge, MA, 02139, 1980.

[15] T. S. Kaczmarek, R. Bates, and G. Robins. Recent developments in NIKL. In *Proceedings of the National Conference on Artificial Intelligence*, pages 978–985. American Association for Artificial Intelligence, 1986.

[16] G. Lakoff. *Women, Fire and Dangerous Things: What Categories Reveal about the Mind*. University of Chicago Press, Chicago, IL, 1987.

[17] H. J. Levesque. Knowledge representation and reasoning. *Annual Review of Computer Science*, 1:255–287, 1986.

[18] H. J. Levesque. Making believers out of computers. *Artificial Intelligence*, 30(1):81–108, Oct. 1986.

[19] H. J. Levesque and R. J. Brachman. Expressiveness and tractability in knowledge representation and reasoning. *Computational Intelligence*, 3:78–93, 1987.

[20] W. S. Mark. Representation and inference in the CONSUL system. In *Proceedings of the Seventh International Joint Conference on Artificial Intelligence*, pages 375–381, 1981.

[21] D. McAllester, B. Givan, and T. Fatima. Taxonomic syntax for first order logic. In R. J. Brachman, H. J. Levesque, and R. Reiter, editors, *Proceedings of the First International Conference on Principles of Knowledge Representation and Reasoning*, pages 289–300, San Mateo, CA, May 1989. Morgan Kaufmann.

[22] J. McCarthy and P. J. Hayes. Some philosophical problems from the standpoint of artificial intelligence. In B. Meltzer and D. Michie, editors, *Machine Intelligence 4*, pages 463–502. Edinburgh University Press, 1969.

[23] M. Minsky. A framework for representing knowledge. In J. Haugeland, editor, *Mind Design*, pages 95–128. MIT Press, Cambridge, MA, 1981.

[24] S. Minton. *Learning effective search control knowledge: an explanation-based approach.* PhD thesis, Computer Science Department, Carnegie Mellon University, Pittsburgh, Pennsylvania, 1988.

[25] J. D. Moore. NIKL workshop summary, 15-16 July 1986. Summary of topics discussed at the meeting among NIKL users and researchers, held at MIT, Sept. 1986.

[26] M. G. Moser. An overview of NIKL, The New Implementation of KL-ONE. Technical Report 5421, Bolt, Beranek and Newman, Inc., 1983.

[27] B. Nebel. Computational complexity of terminological reasoning in BACK. *Artificial Intelligence*, 34:371–383, 1988.

[28] B. Nebel. Terminological reasoning is inherently intractable. *Artificial Intelligence*, 43(2):235–249, May 1990.

[29] A. Newell. The knowledge level. *Artificial Intelligence*, 18(1):87–127, 1982.

[30] P. F. Patel-Schneider. Small can be beautiful in knowledge representation. In *Proceedings of the IEEE Workshop on Principles of Knowledge-Based Systems*, pages 11–16, 1984.

[31] P. F. Patel-Schneider. A hybrid, decidable, logic-based knowledge representation system. *Computational Intelligence*, 3:64–77, 1987.

[32] P. F. Patel-Schneider. Undecidability of subsumption in NIKL. *Artificial Intelligence*, 39(2):263–272, June 1989.

[33] R. S. Patil. Causal representation of patient illness for electrolyte and acid-base diagnosis. TR 267, Massachusetts Institute of Technology, Laboratory for Computer Science, 545 Technology Square, Cambridge, MA, 02139, Oct. 1981.

[34] W. V. O. Quine. Two dogmas of empiricism. In *From a Logical Point of View: Logico-Philosophical Essays*, chapter II, pages 20–46. Harper and Row, New York, second edition, 1953.

[35] C. Rich. The layered architecture of a system for reasoning about programs. In *Proceedings of the Ninth International Joint Conference on Artificial Intelligence*, 1985.

[36] G. Robins. The NIKL manual. Technical report (draft), The Knowledge Representation Project, Information Sciences Institute, 4676 Admiralty Way, Marina del Rey, California 90291, 1986.

[37] J. G. Schmolze and T. A. Lipkis. Classification in the KL-ONE knowledge representation system. In *Proceedings of the Eighth International Joint Conference on Artificial Intelligence*, pages 330–332, 1983.

[38] B. C. Smith. Reflection and semantics in a procedural language. TR MIT/LCS/TR-272, Massachusetts Institute of Technology, Laboratory for Computer Science, 545 Technology Square, Cambridge, MA, 02139, 1982.

[39] S. W. Smoliar and W. Swartout. A report from the frontiers of knowledge representation. Technical Report ISI/RR-90-283, Information Sciences Institute, Marina del Rey, CA, 1990.

[40] W. R. Swartout. XPLAIN: A system for creating and explaining expert consulting programs. *Artificial Intelligence*, 21:285–325, 1983.

[41] M. B. Vilain. The restricted language architecture of a hybrid representation system. In *Proceedings of the Ninth International Joint Conference on Artificial Intelligence*, pages 547–551, 1985.

[42] K. v. Luck, B. Nebel, C. Peltason, and A. Schmiedel. The anatomy of the BACK system. Technical report, Technische Universitat Berlin, Sekr. FR 5-8, Franklinstrasse 28/29, 1000 Berlin 10, Federal Republic of Germany, 1987.

[43] M. P. Wellman. *Formulation of Tradeoffs in Planning Under Uncertainty*. PhD thesis, Massachusetts Institute of Technology, July 1988.

[44] W. A. Woods and R. J. Brachman. Research in natural language understanding. Technical report no. 3742, Bolt Beranek and Newman Inc., January 1978.

Contents