Intensional Subsumption in a General Taxonomic Knowledge Representation Language

by

Scott Thomas Hofmeister

Submitted to the Department of Electrical Engineering and Computer Science
in partial fulfillment of the requirements for the degree of

Master of Science

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

February 1994

Author ....
Department of Electrical Engineering and Computer Science
January 13, 1994

Certified by ....
Jon Doyle
Principal Research Scientist
Thesis Supervisor

Accepted by ....
Frederic R. Morgenthaler
Chairman, Departmental Committee on Graduate Students

Intensional Subsumption in a General Taxonomic Knowledge Representation Language

by

Scott Thomas Hofmeister

Submitted to the Department of Electrical Engineering and Computer Science
on January 13, 1994, in partial fulfillment of the
requirements for the degree of
Master of Science

Abstract

Research during the past decade in the area of KL-ONE style general purpose knowledge representation has encountered some significant empirical and theoretical roadblocks which seem to suggest that the problems will not be amenable to cosmetic solutions. To a large degree these problems are a result of giving extensional semantics to KL-ONE style concepts, and the widely accepted constraint that KR systems should be sound and complete in worst case polynomial time. In a 1990 technical report William Woods sketches the form of a significant alteration in the foundational understanding of concepts and subsumption, along with a number of other important suggestions. In this work we specify a KR language with a set of formal foundations which supports many of the features that Woods advocated.

Thesis Supervisor: Jon Doyle
Title: Principal Research Scientist

# Acknowledgments

I would like to thank:

- Jon Doyle, my thesis supervisor, for his many careful critiques of this work and long hours spent reading drafts. I am deeply endebted to him for enlightening me about many of the subtle points of this area of reserach.
- Peter Szolovits, for welcoming me into the Clinical Decision Making Group and supporting me there. I have appreciated his insightful comments and his willingness to share his knowledge and experience about many things.
- William Woods for the work and ideas that this thesis is built upon, and for the substantial amount of time that he spent with me personally helping me to understand his work, and shaping this project in its early stages.
- Peter Norvig for his early ideas regarding the syntax of the language.
- Isaac Kohane for his advice about how to manage this thesis, for helping me understand the medical domain that this language was initially applied to, and for his warm personal support and concern for my timely completion.
- Ira Haimowitz for introducing me to the research group and helping me in the early stages to understand the research process and the area of general purpose knowledge representation.
- My "family" at 3 Houston Park, Charles Isbell, Bhavya Lal, and Lukas Ruecker, for their unfailing friendship and support which have been so important these last three years, and who have made me feel at home at MIT and in Boston.
- My parents, most of all, for being my most supportive and dependable companions, and for teaching me from day one about the quest for excellence and the importance of hard work.

expressivity, which was the primary deficiency identified by the NIKL/ABEL group [8]. To a significant degree research of the late 80's has investigated the tractability of specific constructs because those have provided clear publishable results. It is our concern that this effort resembles the fable of looking underneath the lamppost because that is where the light is brightest while in fact the bulk of the evidence points towards the conclusion that the real answers are far from the neat and tidy domain of sound, complete and efficient systems.

1.3 Wood's Proposal: Issues and Pragmatics

As mentioned earlier this work is a continuation of the work in Woods' 1990 technical report [31]. In that report Woods advances a number of ideas for the next generation of KL-ONE-like languages. Many of those ideas we have already touched upon through our discussion of representational issues, although some of ideas address pragmatic issues in the construction of KR languages that we have not yet covered. In this section we summarize Woods' proposal, reviewing material from the previous section as necessary.

One way to present the content of Woods' proposal, as well as to motivate it is to detail his framework using his claims about improved performance as the road map. Woods claims that his system provides the following features:

- An *intensional* foundation for conceptual structures, rather then the *extensional* foundations common in languages rooted in first order logic;
- A notion of intensional or *structural* subsumption that is both well-defined and computationally tractable;
- A different metric for the evaluation of tractability results;
- Quantificational tags allowing links to carry explicit quantificational information;
- A mechanism for expressing and combining both definitional and assertional information;
- Means for dealing with multiple equivalent descriptions of a given concept;

- The incorporation of probabilistic information and default values within the framework of structural subsumption; and
- Means for having concepts be simultaneously both generic and individual.

In brief, there are two key central contributions of Wood's work, the first is a concise push away from first order logic and towards the notion of conceptual descriptions and intensional subsumption; and the second is a fairly well developed notion of quantificational tags and how they would be integrated within the framework of intensional subsumption. In the following subsection we will cover the above claims in more detail.

1.3.1 Results of Subsumption and Conceptual Descriptions

Woods advocates significant changes in both the intuitions and algorithms governing subsumption and conceptual structures. In the next chapter we will discuss in more detail the notion of a conceptual description, and provide the formal semantics for intensional vs. extensional semantics. At this point we highlight several of the major benefits resulting from these changes: tractable subsumption, capacity for multiple definitions and partial definitions, and the ability to view concepts simultaneously as individual and generic.

Tractable Subsumption

Although the idea of subsumption appears simple and straightforward, Woods argues in [31] that in practice the basic notion often becomes confused or complicated. He argues that part of the difficulty is that there are really several distinct notions of subsumption and that the word "subsumption" has become overloaded. We now present the notions of subsumption that Woods identified and we will revisit these notions from a more formal perspective in the next chapter:

would stay relatively constant after a certain complexity level is reached.

1.2.3 Definitional and Assertional Representations

The distinction between assertional and definitional information is an issue that the KR community has long struggled with. In 1975 Woods [29] introduced the notions of structural and assertional links, where structural links are those that are used for setting up parts of a proposition or description while assertional links make statements about the world. In response to later systems which were still confused about whether representations were meant to define prototypes or to assert facts, Brachman, Fikes and Levesque in 1983 [4] advanced a number of tenets regarding definitional and assertional information. Doyle and Patil summarized these tenets as follows [6, p. 264]:

1. Definitions define terms.
2. Assertions use terms to state propositions.
3. Storing definitions and assertions separately clarifies the meaning 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.

The response to this approach, as with Brachman and Levesque's stand on soundness and completeness, is that the goal of a clear distinction between terminological and assertional information is a worthy one, but that the position they take is too extreme and hinders the development of practically useful systems.

Consider the difficult experience the NIKL/ABEL group had translating the ABEL knowledge base into NIKL due at least in part to the fact that the NIKL system provided no mechanism for representing or reasoning about individuals. Although it should not be a surprise that this inability to represent instances proved to be problematic, what did turn out to be a notable surprise was the extent to which definitional and assertional information were revealed as being tied together. For example the concepts of treatable and untreatable diseases seem straightforward to define; however the information about whether a particular disease is treatable is usually viewed as assertional. This means that the classifier would not be able to recognize the class of treatable diseases, and following similar reasoning we can conclude the same about, "the concepts of solvable problems, childless persons, feasible schedules, winnable positions, illegal acts, constitutional laws, industrial nations, and democratic countries." [6, p. 282] Others have also noted difficulty of segregating assertional and definitional information [13] and Schmolze and Mark [25] note that the ABEL group was able to mix definitional and assertional knowledge without running into any dire repercussions.

Another approach is presented by Alchourrón, Gärdenfors and Makinson in [1] in which they advocate a system of *epistemic entrenchment* where beliefs are ordered by the strength of the user's faith in those beliefs. Thus one can view "definitions" as lying at the far end of this scale where for any given concept they would be the last thing to relax in the face of an apparent contradiction. Along these lines Woods [29] advocates a framework in which users would be able to distinguish definitions from facts which happen to correlate with a given concept, however the classifier would have equal access to both forms of knowledge.

1.2.4 Concepts and Extensional Meanings

The traditional way to assign meaning to concepts and roles in the KL-ONE family has been to view them as stand-ins for the set of entities or pairs of entities that they represent. Hence the meaning of the concept **Cat** is the set of all the cats in the domain, and thus the meaning of a concept is its *extension.* Although this seems like a reasonable basis for assigning meaning, there are a couple subtle problems that arise.

The first problem is that assertions about concepts whose extension is the empty

Contents

Chapter 1

Introduction

One of the key tasks in the efforts to design "intelligent" automated reasoning systems is that of representing a substantial body of knowledge about a given subject. However, the issue of what is meant by the term *knowledge representation* is complex since every functional application, spreadsheet, and word processing document must contain or encode some form of knowledge. We focus on representation of knowledge in the sense that a KR system must exhibit the following properties:

- One should be able to isolate meaningful structures or statements in the system (this is not usually true for example of neural networks).
- These statements should unambiguously say what the world would have to be like in order for them to be true.
- The system should be able to draw some subset of the conclusions that follow logically from the knowledge represented.
- Ideally the representational framework should be fully general in the sense of being able to represent knowledge from radically different domains but failing that, if it is designed to handle a specific domain it must be able to represent virtually all the relevant information in that domain and it should be somewhat successful in representing knowledge from related domains.

One approach to the problem of representing knowledge is to tackle it as part of a larger system focused on a specific task. Many early AI applications, especially expert systems, incorporated a fair amount of human knowledge, but the mechanisms were often entirely ad-hoc and non-uniform and the knowledge was sometimes represented in program parameter values whose individual semantics were at best a mystery and potentially nonexistent, as with many neural net applications. A different approach involves developing a general purpose domain-independent scheme to explicitly represent wide areas of knowledge. It is upon this area that our work builds.

The area of general purpose KR research began in the late 1960's with the development of semantic networks. Although there was a significant amount of provocative work done in this area, by the end of the 1970's there was a fairly solid consensus that semantic network systems left much to be desired. Although the systems loosely satisfied the definition of KR mentioned above, the semantics were still largely ad-hoc and the meanings of numerous constructs depended solely on the intuitions of the reader, aided only by suggestive naming conventions.

In response, during the late 1970's and the first couple years of the 1980's the KL-ONE system was created [5, 24]. The purpose of the KL-ONE system was to provide a general-purpose domain-independent representation system within an object or concept centered framework. The description of KL-ONE provided clear semantics based on first order logic for concepts and subsumption. However, due to limited resources and a number of unspecified details in the description of KL-ONE, the full system was never implemented and it was NIKL[1] (the immediate successor of KL-ONE [10, 15, 14, 21, 25, 28]) which was much more widely distributed and had the most significant impact on researchers' approach and opinion of this line of research.

The introduction of KL-ONE and NIKL generated considerable interest and activity with the research community. The result of these investigations were some exciting successes with NIKL for applications in very limited sections of natural language understanding, automatic programming and planning. However, efforts to apply NIKL to other tasks ran into a number of substantial obstacles because of the constraints that the NIKL system placed on expressivity. At roughly the same time that this work was being pursued, there was a flurry of tractability research [16, 17, 18, 19, 23] whose impact was that later KR systems became much more focused on addressing

[1]New Implementation of KL-ONE.

these tractability concerns than addressing the issues raised by the users community.

There have been a number of voices advocating a different direction for KL-ONE style research than the one pursued in the late 80's. One of the more prominent has been William Woods, who published an article in 1990 [31] advancing several important ideas for the reform of general purpose KR and KL-ONE style languages in specific. The focus of this thesis involves expanding on this work principally by filling in many of the important details left unspecified by Woods. In the remainder of this introduction we provide some background needed to situate Woods' proposal, and after describing his ideas in some detail we will address our intended contribution.

1.1 Overview of KL-ONE

This section provides a very brief overview of the KL-ONE system, which facilitates creation of a broad network of "structured" concepts standing in specified relationships to each other. The KL-ONE system replaces the unstructured nodes and links of semantics networks with *concepts* which are related to each other through a collections of *roles* and the entire system is organized in a taxonomic hierarchy of generality. Thus concepts are defined through the roles that are attached to them and their placement in the taxonomy.

1.1.1 Roles

Roles were intended to represent any arbitrary relation between concepts, and were viewed as being attached to particular concepts and thus in a sense functioning as attributes of those concepts. For example, the concept of a person could clearly have many roles: place of birth, father, mother, spouse, children, occupation, etc. Since the concept of a person is a generic concept the meaning of these roles would be such that for any given instance of the person concept (i.e. for a specific person) there may exist some instance of another concept which represents this person's child, spouse, etc. The instances which would be the targets of such relations are called the *fillers* of the roles which they satisfy.

Although it is in fact saying something to state that for a given instance of a concept that there might exist an instance of some other concept where a specified relationship holds between the two concepts, in general we want to be more specific and roles provide two mechanism for doing this. The first is through what is called a *value restriction*, which allows the user to require the fillers of a given role to be instances of a specific concept. A trivial example of this is the value restriction that would be placed on the child role of the person concept, restricting the fillers of that role to be people also (i.e. the offspring of humans much be human also). The second constraint on roles is that of *number restrictions* which allows the user to specify the number of fillers of a given role. Thus the father and mother roles would be required to have exactly one filler while the spouse role could have either zero or one.

Finally, KL-ONE provides one other means for placing constraints on the fillers of roles and that was through the use of *structural conditions* (also known as structural descriptions). Structural conditions are attached like roles to a given concept and they allow one to state that a specific relationship holds between the fillers of a given role or between the fillers of different roles. For example in the classic AI example of the blocks world arch, there would be a "lintel" and an "uprights" role attached to the arch concept, both value restricted to being blocks. We would use structural conditions to allow the statement of the constraint that the fillers of the uprights role can not be touching and to state that the fillers of the upright role must support the filler of the lintel role.

1.1.2 Taxonomic Links

The other key component used to represent relationships between concepts is that of taxonomic links. The most common links in the average KL-ONE knowledge base consist of two primary links, those linking roles and concepts which represent value restrictions, and links between concepts which represent taxonomic links. Taxonomic links are used to specify a subclass/superclass relationship between generic concepts. For example, there could be a taxonomic link between the concept of human and that of mammal indicating that humans are a kind of mammal, and similarly there could be a link between the concept of an animal and that of a mammal which would encode the fact that mammals are a specific kind of animal.

As in a frame system, more specific concepts automatically inherit the properties or roles of all the subsuming concepts. However, unlike in a frame system the roles of the subsuming concepts are not treated as defaults in the sense that they can *not* be overridden at a lower level. In general the reason for the creation of a more specific concept is either to tighten the value or number restriction on a given role or to specify that the new concept represents the conjunction of several concepts. For example, we could define the concept of a bachelor by specializing the concept of a man and requiring there to be exactly zero fillers of the spouse role. An example of specialization through conjunction of concepts could involve the definition of the concept of email which would be a subclass both of the concept of a letter and of the concept of an electronic communication.

A major component of KL-ONE which manages the taxonomic links is called the *classifier*. When a new concept is defined, KL-ONE uses the classifier to attempt to determine the appropriate position of the new concept within the taxonomic hierarchy. The most naive algorithm for doing this involves starting at the top of the hierarchy and doing a breadth-first search to test for *subsumption*. One concept is determined to subsume another concept if and only if the first concept is strictly more general than the latter. The classification process is complete after the set of the most specific concepts which all subsume the new concept has been found and all of concepts in the set have been explicitly asserted to subsume the new concept unless those links are already part of the concept definition. The nature of this set of most specific concepts is such that no concept in the set should be subsumed by another other concept in the set, and for any concept **X** in the taxonomy which is not in the set but which subsumes the new concept, there must a concept in the set which is a specialization of **X**.

Finally, KL-ONE also permits the construction of concepts with only partial definitions through what are called *primitive* concepts. Primitive concepts were intended to represent what philosophers termed "natural kinds", meaning concepts such as "bird" or "hand" where no complete definition is feasible since it is possible to continually propose borderline cases (e.g. penguins, paws) which challenge almost any rigid definition. For non-primitive concepts in KL-ONE a definition specifies a set of necessary and sufficient conditions, and for primitive concepts a definition contains only necessary conditions. Thus the classification algorithm will never place a newly defined concept as an instance or subclass of a primitive concept, however if it explicitly stated in the concept definition that the new concept is an instance or subclass of a primitive concept then it will automatically inherit all the roles that are attached to that concept.

1.2 Representation Issues

In the development of successors to KL-ONE during the second half of the 1980's, the discussion about how to construct such descendants was characterized by a vigorous debate involving several important issues. To a large degree, these issues were, and still are, taken to be the most important axes along which general purpose KR languages are measured. The issues are:

Soundness: Given a set of sentences Γ, a sentence ϕ is *logically entailed* (written $\Gamma \models \phi$) if and only if every interpretation and variable assignment that satisfies the sentences in Γ also satisfies ϕ. In other words, $\Gamma \models \phi$ if and only if $\models_I \Gamma_{[U]}$ implies $\models_I \phi_{[U]}$ for all interpretations I and variable assignments U, where an interpretation maps the constant symbols in the language to elements of a conceptualization and a variable assignment maps variables to elements of the conceptualization. We say that an inference procedure is *sound* if any sentence that can be derived from a database (i.e. Γ) using that procedure is logically entailed by that database. The main manner in which this issue surfaces in KR systems involves the discussion of whether to allow default information and the associated inferences which are not sound.

Completeness: We say that an inference procedure is *complete* if and only if any sentence that is logically entailed by a database can be derived using that procedure. Developing KR systems with sounds and complete inference procedures is in general very difficult.

Expressivity: A definition of global expressivity is difficult to provide, so we will instead give a relative definition by saying that a given system is more expressive than another system if it can represent everything that that system can represent and some set of relationships that it can not. As will be discussed in more detail shortly, questions in many languages are undecidable, and even in relatively constrained languages, inference procedures that are sound and complete may require exponential time for certain tasks. Thus positions on this issue generally involve statements about what kind of expressivity is most crucial and what form of constraints on it are acceptable.

Efficiency: By efficiency we mean the running time of the representation system on a given task. The question of how to measure the efficiency of general purpose KR systems is as crucial as it is contested. Part of the debate involves whether we should be focusing on the complexity of classification or the complexity of subsumption, and another part of the debate involves the question of whether the worst case time complexity should be required to be polynomial, exponential, or simply decidable.

Assertional Information: The issue of the how or whether to separate definitional and assertional information has turned out to be a vexing problem for many general KR systems. A broad spectrum of researchers have advocated a variety of positions on this issue, ranging from forbidding any assertional information, to allowing it to be mixed in indiscriminately with definitional information.

Extensionality: There has been considerable debate around the subject of whether first order logic under the standard extensional interpretation of concepts and roles is an appropriate base representation for KL-ONE-like languages.

A large number of papers and systems have addressed the issues above, but we will discuss only a few of them. We will begin our discussion with a seminal paper [29] that William Woods published in 1975 in response to mainstream of semantic networks research. This paper argued for the development of a clear set of explicit semantics in future systems and strongly pushed people to provide a better characterization of the constructs and inferences supplied by their systems. This philosophy was further extended and reached its most extreme in a highly influential paper published in 1984 by Ronald Brachman and Hector Levesque [11]. In this paper Brachman and Levesque essentially argued that KR systems should provide sound and complete inference systems, and that such systems should have at worst polynomial time complexity in the worst case. After some time many researchers came to the conclusion that the constraints that Brachman and Levesque had specified, although well motivated, were placing a crushing burden on the developers of KL-ONE-like systems if they aspired to be at all useful. In 1989, Doyle and Patil published a lengthy rebuttal [6] to Brachman and Levesque's original paper in which they argued for more expressivity, a relaxation of the emphasis on completeness, and advanced the notion of "rational" management of inference tools. Much of this work was based in the experience of a group at MIT which attempted to represent the ABEL knowledge base [20] in NIKL and identified some important expressivity demands that NIKL could not satisfy. Finally, in 1990 William Woods [31] published a technical report which provided the outline of a system which was had at its foundation a series of positions on the above issues which departed notable from Brachman and Levesque's proposal in an effort to provide for a more expressive and therefore more practically useful KR systems.

We will now catalogue the positions that the above players have taken on the representation issues that we have discussed above:

1.2.1 Sound and Complete Inference

The debate surrounding the value of sound and complete inference procedures, and specifically complete inference procedures has in many ways had the most significant impact on this area. Brachman and Levesque advocated that all inference procedures for KL-ONE-like languages should be sound, meaning that a false conclusion can never be drawn, and also complete. What they meant by complete was that the system should be able to determine "efficiently" whether a given proposition is entailed by the information in the current knowledge base. More specifically, the system needs to be able to recognize equivalent definitions despite different phrasings or syntactic constructions. For example, the system would be expected to discover whether P=NP provided that the concepts of the problems in P and problems in NP could both be fully represented.

In [12, p. 79] Levesque and Brachman motivate this demand for completeness by invoking the goal of a *shared understanding* which means that the system understand what the knowledge means to user. To a large degree this is based on Smith's [26] argument that the system need not 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 behavior as if it believed these propositions."

In contrast Doyle and Patil argue that this demand that definitions be fully understood by the classifier before they can be used at all is way too strong. They argue that it is possible to share understandings of meanings even if those meanings are not 100% complete. Doyle and Patil draw a parallel to the common distinction between "knowing" and "understanding", and note that there is a wide gulf between these two terms. As many science teachers are well aware, it is possible for a student to "know" facts or formulas, but to still not be able to understand them. Thus a more reasonable demand would be that the user and the representation system have shared knowledge but not necessarily a shared understanding.

On the soundness issue, both Woods and Doyle and Patil advocate the inclusion of default information, and thus unsound inference into the framework of a general KR system. Although, the KR community had previously viewed the areas of default reasoning and structural subsumption as being orthogonal if not contradictory, the integration of these two approaches is now an area of active investigation.

1.2.2 Tractability Analysis

Shortly after the development of the NIKL system there were a flurry of tractability results which showed that subsumption in even moderately expressive systems is of exponential complexity in the worst case when it is even decidable. In their paper Brachman and Levesque presented a simple example of a language that implemented a very limited subset of FOL, and they demonstrated how the addition of a fairly simple construct caused the subsumption procedure to go from being computable in $O(n^2)$ time to being undecidable. Brachman and Levesque urged the research community to focus its energies on mapping out the boundary between intractable and tractable subsets of FOL and especially to, "discover the crossover points where small changes in a language change its computational character completely."[11, p. 67]. This question of analyzing the different sets of FOL and determining the crossover points proved to be a tantalizingly crisp problem description which received a significant amount of communities' attention during the 1980's.

Brachman and Levesque's argument that the proper metric for evaluating tractability results was the worst-case analysis was motivated by the existence of some critical applications for which an unacceptable worst case behavior would make the system unusable. Although they admit the practical importance of the average case scenario over the worst case analysis, they observe that there had been very little success in formally characterizing average case behavior and that it would be, "irresponsible" for computer scientists to be "providing general inferential service if all that we can say about it is that by and large it will probably work satisfactorily." Although they never argue explicitly for it, it is clear that acceptable behavior meant that the complexity should be some small polynomial.

Doyle and Patil take issue with this strong position and argue that there are very few standard artificial intelligence applications in which a failure on a routine task has horrible consequences. In addition they observe that in real applications tolerance for the occasional late or missing answer is quite common, and that the appropriate model in which to view an inference mechanism is one in which the progress of the reasoner is periodically reviewed and potentially aborted by some form of metareasoning system. Furthermore, although it may be difficult to clearly characterize the mathematics of the average case they argue that as computer scientists we should be able to use, without embarrassment, summaries of real-world practical experience in estimates of average running times.

Lastly, Woods in [31] takes issue with the other component of previous tractability research, namely that these results have addressed the question of complexity in terms of the complexity of the subsumption rather than in terms of the complexity of classification. He argues that in most application the size of an individual concept description is bounded by some reasonably small quantity, and thus determining subsumption between any two concepts can be viewed for most practical purposes as a constant time operation. Thus it is the cost of classification which comes to dominate as the knowledge base grows, rather than the cost of subsumption which set (e.g. unicorns, prime numbers less than 1, and confirmed cases of the transmission of AIDS through casual contact), assertions involving those concepts sometimes have no semantic content. For example the statement "all unicorns drive trucks" is vacuously true if there are no unicorns in the domain to quantify over. The other problem is that although two concepts which have the same extension by definition have the same meaning, there are an abundance of examples where this violates our intuition. For example the concepts of morning star and the evening star are equated and even more egregious is that so would the concept of a unicorn and the concept of a confirmed case of the transmission of AIDS through casual contact. Although knowing that two concepts have the same extension may be a valuable piece of information, we do not want it to be the dominant relationship linking the two concepts and we certainly want to be able to continue to distinguish the two entities (something which none of the KL-ONE descendants allow you to do).

Finally, it is worth noting that the weighty demands of completeness, that definitions be recognized as equivalent regardless of different phrasings or syntactic constructions, is due to the extensional nature of subsumption. Woods postulated in [31] that there are several distinct notions of subsumption and that the common usage of the word has become overloaded. He argues that in most of the KL-ONE family the concept of extensional subsumption has been used this leads to unrealistic goals of completeness (e.g. proving P=NP). However, he also states that it is possible to formalize a less strict version of subsumption (intensional subsumption), which allows for the goals of soundness and completeness in polynomial time.

1.2.5 Real KR Expressivity Needs

Lastly, in this section we outline some of the expressivity needs that have been identified by some of the broader research efforts that used NIKL. Unfortunately, there have been very few research efforts to apply KL-ONE-like technologies to complex real world problems. The two principal efforts involved one lead by Haimowitz, Patil and Szolovits [8] and the other by Smoliar and Swartout [27]. Both these projects found the language restrictions in NIKL to present very serious obstacles to the development of useful knowledge bases. In [8] Haimowitz describes the experience of trying to translate the preexisting medical knowledge of ABEL [20], a program for the diagnosis of acid-base and electrolyte disorders, into NIKL. The following were some of the key short-comings of the NIKL system encountered in that effort:

- No assertional component. This prevented the creation of instances of a given concept, and forced all statements made about generic concepts to be definitional rather than assertional.

- Haimowitz *et al.* claimed that NIKL was unable to represent several key relationships including the part-whole, containment and other spatial relationships, and causation. Although, Schmolze and Mark latter showed [25, pp. 64-66] that it is possible to represent the part-whole relationship in NIKL, their solution admittedly does not address the complaint that the part-whole representation should be assertional rather than terminological. They also concede that their representation is potentially cumbersome and unintuitive.

- An unrealistic requirement that all concepts have a complete list of necessary and sufficient conditions in order not to be treated as primitive concepts and thus essentially ignored by the classifier.

- No ability to reason with numeric intervals, sequences, sets, and temporally varying quantities.

- No capacity for multiple definitions (i.e. multiple sets of sufficiency conditions for a given concept, the satisfaction of any one of which signals the satisfaction of the concept), and no ability to provide multiple names (i.e. synonyms) for a given concept.

- A counterintuitive if not incorrect implementation of statements of disjunction and set exhaustion.

Although some of these issues have been dealt with in latter descendants of NIKL, largely there has been little work within this family of languages towards increased

Extensional Subsumption: This means that any instance of the subsumed concept must be an instance of the subsuming concept in the *model-theoretic* sense.[2] Another way to state this is that the extension of the subsuming concept is a superset of the extension of the subsumed concept.[3]

Axiomatic Subsumption: A more general concept is asserted to subsume a more specific concept as a direct result of an axiom of the knowledge base. For example the interval (**at-least** 5) subsumes the interval (**at-least** 6).

Recorded Subsumption: Involves the case where a more general concept is explicitly recorded to subsume a more specific structure either through a direct link or through the transitive closure of such link.

Structural Subsumption: A concept would structurally subsume another concept if it is determined to be more general than the subsumed concept as the result of some formally specified subsumption criterion applied to the structures of the descriptions. The algorithm for determining structural subsumption essentially rests on the ground set of axiomatic subsumption relations. The key difference between structural and axiomatic subsumption is that axiomatic subsumption governs only subsumption between concept components where structural subsumption combines these axioms and their relations to each other to determine subsumption between complex concepts.

All of the KL-ONE family have subscribed to the notion of extensional subsumption in large part because it is both mathematically and intuitively very clear. However as we have discussed already there are some significant drawbacks to this interpretation. For example, determining whether the extension of one concept contains the extension of another concept is often undecidable not to mention non-polynomial, and simply because two concepts have the same extension does not mean that they should be equated. In putting forward his notion of structural subsumption Woods does not propose a notably different algorithm for computing subsumption, rather he proposes a different lens through which to see concepts and subsumption.

At the heart of the idea of intensional or structural subsumption is the intuition that a concept should subsume another concept only if this relation can be derived from inspection of the "syntax" or "structure" of the concept and that a more general

[2] By a model theory we refer back to the functions U and I from section 1.2 which map symbols in the representation language to elements in the conceptualization.

[3] In this presentation we merge Woods' concept of "deduced subsumption" and "extensional subsumption".

deductive system would be charged with drawing more domain dependent inferences. Thus for example while the concept of a polygon with three sides would extensionally subsume the concept of a polygon with three angles, there is no direct structural relationship that would allow this inference to be drawn. However, a structural subsumption procedure would determine that the concept of a polygon with at least three sides would subsume the concept of a polygon with exactly three sides.

Thus as Wood's states [31], "if subsumption is defined by a structural relationship rather than a model-theoretic criterion, it can be defined to have a tractable subsumption computation — i.e. one can choose a structural subsumption relationship with a tractable computation as the definition."

Multiple and Partial Definitions

The KL-ONE system allowed neither multiple definitions of a concept, nor the ability to allow for a partially defined concept. However, both of these issues were revealed as being crucial issues for medical representation through the work of the NIKL/ABEL group. Often a concept may have more than one set of sufficient conditions for example the term "acidemia" may be defined as decreased pH, or similarly as an increase in the hydrogen ion concentration [7]. Ideally we would like the system to be able to record these multiple definitions and to recognize a new concept as satisfying the defined concept if *any one* of its definitions is satisfied.

Partial definitions in KL-ONE were excluded because the system required a set of necessary and sufficient conditions for any concept definition. However, within the field of medicine as well as other domains, the state of human knowledge is quite incomplete and specifying necessary and sufficient conditions for every concept, or even most concepts is impossible. In KL-ONE and NIKL any concept for which it was not possible to encode a set of necessary and sufficient conditions was marked as primitive and thus essentially ignored by the classifier.

Woods suggests that in order to determine if concept A subsumes concept B you first find the conjunction of the necessary conditions attached to B and then query whether any one of the sufficiency sets attached to A is satisfied by the conjunctions of B's necessary conditions. Effectively this allows for multiple definitions since more than one sufficiency set can be given, and it allows for partial definitions since it is possible to specify either only necessary or only sufficient conditions for a given concept.

Individual Concepts

In keeping with the intensional view of concepts, Woods argues that there should no longer be a distinction between generic and individual concepts, or rather that it should be possible to view a concept simultaneously as both generic and individual. Hence both traditional "individuals" (e.g. **Ronald-Reagan**) and "generic concepts" (e.g. **government-official**) would be represented using conceptual descriptions, and it would be possible for any concept to be considered an instance of any other concept. An example of where this would be useful involves the concept of a particular doctor, **Louis-Pasteur** which can be an instance of the concept of a **doctor** and that concept in turn could be an instance of the concept of a **medical-professional**.

1.3.2 Quantificational Tags

Woods in a personal communication described his system of quantificational tags as the most important contribution of his technical report. Woods claims that much of the early confusion in the semantic networks community had revolved around relying on the informal semantics of the names of the nodes and links. For example consider the triple, (**doctor** *works-in* **hospital**). The English translation of this sentence is that doctors work in hospitals, or more formally that $\forall x \in \textbf{doctor}, \exists y \in \textbf{hospital}$ such that (x *works-in* y). However, it is easy for such a statement to have hidden semantics, for example do all doctors work in hospitals? Is a person still a doctor if s/he does not work in a hospital? Is the place where a doctor works automatically a hospital, and is a building still a hospital if no doctors work there?

Although few would debate the sentence, "doctors work in hospitals", when you pin down the semantics as specified above many of the logical consequences provide answers to questions we have mentioned which are undesirable. The moral is that if certain important distinctions are not made in the representation, many unintended consequences will result. To bring these distinctions to the fore Woods proposed a system of quantification tags which he conceived of as relational operators. For example the operator AA could be applied to a relation to state that every instance of a given class stands in a specified relation to every instance of another class (e.g. (republican ([AA] voted-for) republican-candidate) would assert that every republican voted for every republican candidate), and the tag EE denotes that there exists some member of the first class who stands in the specified relation with at least one member of the second class (e.g. (democrat ([EE] voted-for) democratic-candidate) means that some democrat voted for some democratic candidate).

1.3.3 Assertional vs. Definitional vs. Defeasible Values

As discussed earlier, the distinction between assertional and definitional information has been problematic since the early days of semantic nets. In addition, the capacity to record defeasible values has been at the core of many frame-based reasoning systems, and is clearly crucial for a significant number of applications. Originally, in the days of semantic nets no distinction was made between assertional and definitionally information and then in much later systems such as KRYPTON the segregation was strictly enforced. In this system we follow the results of the NIKL/ABEL group which found the strong need to mix definitional and assertional information and that mixing them did not cause major negative consequences.

Woods' advocates allowing the three different kinds of knowledge to be mixed in the sense that the classifier would have equal access to all three, however he also contends that it is important to clearly distinguish which information belongs to which camp.

1.4 This Thesis

Unfortunately Woods' technical report provides only a sketch of a system which provides the features he discusses. Although some ideas are developed more fully than others, there are a quite a number of gaps between Woods' proposal and a usable system. This thesis is an effort to bridge those gaps, and although we also leave a number of issues unresolved we attempt to do so in such a way that the core elements of a functional system are in place and that future work can be seen as extensions of this work rather than as a reworking of the foundations. In summary the focus of this work involves providing mathematical semantics for Woods' notions of concepts and intensionality in addition to transplanting his notion of quantificational tags into a more complete framework. The follow subsections mirror the previous section, and for each point we will describe to what extent we address the issue.

1.4.1 Intensional Subsumption

In the previous section we presented Woods' notion of structural subsumption which shifted away from determining subsumption by evaluating the subset inclusion of the extension of concepts and towards a notion of subsumption based solely on the syntactic structure of the concepts. Thus the details of exactly how the structural subsumption algorithm operates becomes increasingly important. Although Woods discusses how structural subsumption should work for a variety of constructs, and although the algorithm for the remaining constructs is not complex, we complete Woods' language definition and specify the subsumption relationship.

1.4.2 Individual Concepts

Woods advocates a radically different perspective on the issue of generic versus individual concepts. This switch necessitates a fundamental shift in the semantics of concepts since many KL-ONE systems' semantics have at their core the distinction between concepts and individuals, where individuals are the elements of the model domain and generic concepts represent sets of individuals. Woods does not address the issue of how the semantics of concepts should be changed, although this thesis provide some further guidance.

1.4.3 Quantificational Tags

In this work we leave the underpinnings of Woods' system of tags largely untouched. Instead of having the tags function as relational operators however, we make the tags required components in certain constructs in the language. In addition we have modified the syntax to increase readability. Some of the ways that the above statement might be reconstructed in our language is by stating ((every **doctor**) *works-in* (some **hospital**)) and ((some **doctor**) *works-in* (every **hospital**)), or ((every **doctor**) *works-in* (only **hospitals**)). Clearly some of these statements are not true, but the advantage of our systems is that it first allows for the distinctions to be made and secondly, in most cases either the intuitive semantics and the formal semantics are both correct or they are both incorrect.

1.4.4 Flavors of Information

Finally, although Woods discusses at some length the importance and difference between assertional, definitional, and defeasible information, he does not discuss how, or at what point the information would be treated differently and we do not address this issue either. It seems that the most obvious use would be for doing very simple resolution of contradictions between statements. The only difference between this work and Woods' is that we propose a somewhat different marking scheme for distinguishing the different kinds of knowledge in an attempt to improve the clarity of the system.

1.5 Plan of this thesis

In this chapter we presented an overview of this work and outlined the history of the research upon which our KR language is built. We have presented a list of claims that

Woods had made about the benefits of his framework, and we have discussed each of these claims in turn and explained how this work has addressed those points. In the remainder of this thesis we provide the intuitive, semantic, and syntactic details of the language.

Chapter 2 describes in detail the syntax the constructs in the language. We will begin with some examples of knowledge represented in the language to provide the reader with a feel of how the entire system operates before plunging into the details of individual constructs.

In the next chapter we present the core contribution of our system, namely the semantic foundations. We provide precise semantics for Woods' somewhat under-specified notion of conceptual descriptions and intentional subsumption, and with situate this with respect to traditional extensional semantics and subsumption. We also detail the meaning of constructs in the language, and we present the rules which govern which subsumption relationships are deducible. We will prove that our inference relation is sound with respect to the semantics and we will postulate that it is also complete.

In the final chapter, we will review Woods' claims and reflect upon how the previous chapters have addressed those claims. In addition, in the discussion of each claim we will talk about the remaining work that needs to be done to fully substantiate the claim. Lastly, we will close with a discussion of the possible future directions of our work.

Chapter 2

Language Syntax

2.1 Preliminary Distinctions

In this chapter we present the syntax and informal semantics of our language. Because of the high degree of mutual recursion of terms in the language we wish at the start to make some clarifying distinctions. The first is the distinction between operations which modify the knowledge base(KB), and expressions which are used to describe the entries in the KB. We also examine the relationship between descriptions, concepts and modifiers, and lastly present some basic examples in order to provide a better sense of the use of the language.

2.1.1 Expressions vs. Commands

Although in this thesis we refer to a knowledge representation *language* and that is indeed where the bulk of the research effort is. What we are really describing is a KR system, the distinction being that a language is centered around "expressions", the construction of concepts, and their interrelations, while a KR system includes all this in addition to a mechanism to store and manage knowledge communicated via the language. In most KR systems, the operations on the KB simply act as database operations, manipulating the elements without regard to the kind of data represented. In our system, some of the commands/operations carry semantic weight and it is those commands that are the focus of our final section. Although from a practical point of view other commands which allow the user to search and display the knowledge base are of high importance, since they are essentially peripheral to the representational issues which are our focus, we leave their enumeration for future work.

2.1.2 Concepts vs. Modifiers

Earlier we introduced the notion of a conceptual description as the key building block of the representation system. We now further refine that notion by discussing several different "subtypes" of conceptual descriptions. While most of these categories are straightforward, the distinction between these two particular categories, concepts and modifiers, merits some additional explanation.

Concepts are used to represent those things *which are most naturally reasoned with as individuals or classes.* For example, Ronald Reagan, the set of all Labrador retrievers, and the citizens of India would all be perfectly reasonable concepts. By contrast, we use modifiers to represent those things *which are most naturally reasoned with as predicates.* Examples of modifiers could include such items as having blond hair, eating fatty foods, being pregnant, or being sunburned. A modifier would be either true or false of given description. Modifiers thus correspond to functions while concepts represent sets.

One can of course identify these two notions through the standard math construction of the characteristic function. Hence they are identical from that view, and indeed our semantics for these two constructs will also be identical. However, we still wish to distinguish these concepts at some level since it seems that a large component of naturalness has been given up in previous KL-ONE-style systems by forcing people to represent everything either as one or the other. For example, it is rare to hear a physician reasoning explicitly about the set of people whose blood sodium level is above 150meq/L (forcing a function into a set), and similarly it would be unnatural to reason with a predicate for doctorness (forcing a set into a function). It is also worth noting that there are clearly some entities which can be comfortably viewed either as concepts(sets) or modifiers(functions), or both (e.g. "solid" means either a solid object or the quality of being solid). Our intent is not to imply that everything can be clearly identified as a concept or a modifier, but to avoid the requirement of shoehorning all things into just one of these forms. Although there is more to be said about the importance of this distinction, we will revisit this topic later in the context of the medical knowledge represented so far.

2.2 Overview and Examples

In table 2.1 we show an overview of the constructs of the language, and in the following we present a few examples to provide a basic feel for this syntax. In the statements below it is helpful to keep in mind that dcl stands for declare, def for define, typ for typically, qty for quantity, and restr for restriction. Note that the convention we follow in presenting statements in the language is that concept names are represented in bold (e.g. **people**), reserved words are represented in san-serif (e.g. constrain), and the names of relations are in italics (e.g. *have-disease*). We follow LISP conventions of identifying symbols and numbers, so the typecase of the language is irrelevant.

Ex. 1 *Magic Johnson is HIV+.*

(dcl **Magic-Johnson** *instance-of* **HIV-positive**)

Ex. 2 *The kidney is a kind of internal organ.*

(def **kidney** *kind-of* **internal-organ**)

Ex. 3 *Every person needs every vitamin.*

(dcl (every **person**) *needs* (every **vitamin**))

Ex. 4 *Every drug is taken by some patient.*

(dcl (every **drug**) (inverse *takes*) (some **patient**))

Ex. 5 *Only people with HIV are Larry's female sexual partners.*

(dcl **Larry** (constrain *sexual-partner* **female**) (only **person-with-HIV**))

Ex. 6 *A person with a fever is a person with temperature over 99F.*

Descriptions		<unpaired-description>
		(<unpaired-description>,<unpaired-description>)
	Interval	(<numeric-interval>,<units>)
Unpaired Descriptions	Concept	<name>
		<built-in-concept>
		(<logical-op> <concept>+)
		(RESTR <concept> <modifier>* {with <stmt>})
	Modifiers	(REL <relation> <quantified-descrip> {(label <name>)})
		(PRED <name>)
	Relation	<name>
		<built-in-relation>
		(<relation-op> <relation> {<concept>})
	Statement	(<quantified-description> [<relation> <quantified-description>]+)
		(<logical-op> <stmt>)
	Numeric Intervals	<num>
		([<numeric-rel-op> <num>]+)
Quantified Descriptions		<description>
		(<tag> <description>)
		(<numeric-tag> <interval>)
Commands	Assertions	(DEF <stmt>)
		(DCL <stmt>)
		(TYP <stmt>)
		(NAME <name> <description>)

Table 2.1: Overview of syntax of language.

(dcl (restr **person** (*have-disease* (a **fever**))) *same-as*

(restr **person** (*temperature* (qty ((> 99),**Degree-Fahrenheit**)))))

As should be clear from the above, the format of the average entry to the KR system is of the form of two descriptions with their attached quantificational information (quantifiers apply left to right), the relation that holds between them, and an instruction at the beginning indicating how the information should be interpreted.

2.3 Descriptions

Descriptions represent the core of this system and the constructs listed in this section represent the top level "expressions" in the language.

```
<description>
            = <unpaired-description>
            | (<unpaired-description,unpaired-description>)

<unpaired-description>
                     = <concept>
                     | <modifier>
                     | <relation>
                     | <interval>
                     | <statement>
                     | <s-expression>
```

Note that we draw a distinction between descriptions and unpaired descriptions because we do not want to inject the additional complexity of having the elements of ordered pairs of descriptions themselves be ordered pairs. The *raison d'être* for ordered pairs as descriptions is to represent intervals and to provide "instances" for relations. For example the statement (**Robert-Kennedy** *brother* **John-Kennedy**) and the statement ((**Robert-Kennedy**,**John-Kennedy**) *instance-of brother*) would have exactly the same semantics.

2.3.1 Statements and Quantification

After descriptions, the next most important construct in the language is the statement. A statement represents a proposition which can be either true or false, and a simple statement consists of an infix binary relation and two quantified descriptions. A compound statement combines statements with the usual boolean operators.

```
<stmt>  = <simple-stmt>
        | <compound-stmt>

<compound-stmt>
        = (OR <stmt>*)
        | (AND <stmt>*)
        | (NOT <stmt>)

<simple-stmt> = (<non-numeric-quantified-description>
                     [<relation> <quantified-description>]+)

<non-numeric-quantified-description>
                                  = <description>
                                  | (EVERY <description>)
                                  | (A <description>)
                                  | (SOME <description>)
                                  | (ONE <description>)
                                  | (NO <description>)
                                  | (ONLY <description>)

<quantified-description>
                        = <description>
                        | (EVERY <description>)
                        | (A <description>)
                        | (SOME <description>)
                        | (ONE <description>)
                        | (NO <description>)
                        | (ONLY <description>)
                        | (NUM <interval>)
                        | (QTY <interval>)
```

The reason for providing the option to have multiple occurrences of `(<relation> <description>)` in simple statements is to allow for multiple statements to be made about a single entity without having to restate the entity name or description multiple times. An example statement would be the following:

(**Bill-Clinton** *hair-color* **silver**
eye-color **blue**
place-of-birth **Hope-Arkansas**
....)

Every simple statement asserts a relationship between two objects, similar to a predicate calculus statement which asserts a relationship between two variables, where the two variables are bound by quantifiers. In our system the quantified concept serves as a means to introduce these quantifiers. Note also that quantified descriptions are not descriptions in their own right and thus have no meaning when used by themselves (just like the expression "only AIDS-patients" has an undefined meaning when used out of context). Quantified concepts serve only to shape the semantics of the statement or modifier in which they are used.

The quantifiers **every** and **some** correspond to the logic quantifiers $\forall$ and $\exists$, thus for example the statement ((**every man**) *loves* (**some woman**)) means that for each man there exists at least one woman whom he loves. The quantifier **one** is equivalent to ι — that is, it means that there will be only one filler of the specified class for that relation. **No** means that no such object can fill the relation, and **only** means that every object that fills the relation must be of the specified type. The tag **a** is provided exclusively to increase readability and has exactly the same meaning as **some**. Finally, **num** and **qty** provide for the use of a range of values, e.g. ((**every Bill-Clinton**) *close-friends* (**num** ((**from** 2 **to** 5),**fillers**)) says that the number of "fillers" of the close-friends relation with Bill Clinton is between two and five, inclusive, we will discuss the construction of intervals shortly. Similarly, (**every truck**) *weighs* (**qty** ((**from** .5 **to** 5),**tons**)) expresses a constraint on the weight of any truck. **Num** would not be appropriate here because each truck has only one weight.

Although the use of statements to make assertions is fairly obvious, the use of statements as "descriptions" is less clear. However, if one views a statement as describing a truth value (in this case either true or false) then clearly it is possible.

The costs and power of having statements be able to be descriptions are not clear at this point, however the benefits of maintaining a uniform building block or core element are apparent.

Finally, note that we have explicitly disallowed numeric tags from being used in the first quantified description in a statement. Although there are some cases where the use of a numeric tag in the first position would make sense (i.e. 5 billion people is the population of the globe, or 100 MIT professors require 2.5 administrators) it is usually possible to rephrase these statements by eliminating the numeric tag from the first position. Furthermore, many expressions which use a numeric tag in the first position are either meaningless or their semantics would be very cumbersome to describe in a general way.

2.3.2 Concepts

As noted previously modifiers and concepts are two key sub-types of descriptions. The structure of a concept is as follows:

```
<concept>
        = <name>
        | <built-in-concept>
        | (AND <concept>+)
        | (OR <concept>+)
        | (NOT <concept>)
        | (RESTR <concept> <modifier>* {with <stmt>})

<built-in-concept> = TOP-CONCEPT | TRUE | FALSE
```

This syntax outlines the basic concept-building operations and the main way to link concepts and modifiers. As can be surmised, the above operators allow for conjunction, disjunction, and negation of concepts. The **restr** essentially restricts a given concept to only those instances which satisfy the constraints detailed by the modifiers and the statement. For example,

```
(restr fatal-illness (rel illness-locus kidneys)
                     (rel risk-population
                          (restr men (rel age (greater-than 50)))))
```

This concept represents the description of a fatal illness of the kidneys for which the major risk group is men over the age of 50.

We also include in the language several built-in concept. The concept **top-concept** represents the most general concept in the hierarchy and the concepts **true** and **false** are descriptions of truth values which we shall use in manipulating the value of statements. At some point we expect to add concepts from the LISP type hierarchy like string and number.

2.3.3 Modifiers

```
<modifier>
        | (REL <relation> <quantified-description> {(label <name>)})
        | (PRED <concept>)
```

Modifiers can be of exactly one of two forms. The first option is that it consists of a relation and a quantified concept. The descriptions which satisfy the modifier are those which stand in the specified relation with the quantified concept. For example, the modifier (rel *takes* (a **math-course**)) can be viewed as a predicate which is true only of descriptions of entities which are taking at least one math class. The concept (restr **student** (rel *takes* (every **math-course**))) describes students who are taking every math class. The other format a modifier can take begins with the reserved word pred and then usually a name, and is used to convert a concept into a predicate or modifier. For example the modifier (pred **red-haired**) would be a description of all red haired entities.

Modifiers also allow for the ability to add a label so that it is possible to refer to the fillers of a given relation in the statement which may appear at the end of a restr clause. In fact the purpose of this optional statement is to assert constraints that must hold between certain fillers of a given concept. For example:

```
(restr student (rel favorite-professor (one Person) (label L1))
               (rel taking (some class) (label L2))
     with ((one L1) teaches (some L2)))
```

This concept describes students whose favorite professor is teaching some class that s/he is taking.

2.3.4 Relations

We next consider relations, which besides being concepts in their own right, have a special place within simple statements. At present there are six ways to refer to a relation: by name, by the name of a pre-defined built-in relation, or by one of four relation-forming operators:

```
<relation>
        = <name>
        | (RELATION <name>)
        | <built-in-relation>
        | (INVERSE <relation>)
        | (ALL <relation>)
        | (CHAIN <relation>*)
        | (CONSTRAIN <relation> <concept>)
```

The operator inverse forms an inverse relation: (**x** (inverse *R*) **y**) holds whenever (y *R* x) holds. The all operator is used to get at the set of fillers of a relation. For example, the fillers of the parent relation for a person will be people, but the unique filler of the (all **parent**) relation will be a set of people (which ordinarily has cardinality 2). The chain operator is used to compose relations; the relation (chain *brother parent*) means uncle. The constrain operator refers to those fillers of a relation that happen to be of a particular type. For example, (constrain *sibling* **female**) has the same meaning as the *sister* relation.

```
<built-in-relation>
        = RELATION
        | SAME-AS
```

```
| SPECIALIZES
| KIND-OF
| SUBSUMES
| SATISFIES
| INSTANCE-OF
```

The list of built-in relations defined above is bound to grow but for now most of the predefined relations involve subsumption. The relation, *relation* is the top-level relation which subsumes all other relations. The relations *kind-of* and *instance-of* were touched on in the introduction and will receive a more detailed treatment in the chapters on semantics. The relation *satisfies* is identical in meaning to the *instance-of* relation although it is intended to be used to state that a given concept satisfies a certain modifier. Similarly, the *specializes* relation is identical to the *kind-of* relation except that it is intended to be used with concepts while *kind-of* is intended to be used with modifiers. The *subsumes* relation is also identical to the *kind-of* and *specializes* relation except that it is intended to maximally general and can be used between any pair of descriptions. The statement (A *same-as* B) is an abbreviated way of stating (A *kind-of* B) and (B *kind-of* A). Note that this is equivalent to asserting a mutual subsumption relationship, which causes most KL-ONE-like systems to view the concepts as identical. However we wish to maintain a distinction between mutual subsumption and identity, and this will be discussed further in the next chapter.

2.3.5 Quantities: Numbers, Intervals

Numbers are represented in our system through the use of descriptions which we call intervals.

```
<interval> = (<numeric-interval>,<units>)

<numeric-interval> = ([<numeric-rel-op> <num>]+)

<numeric-rel-op>
        = <           | >           | =       | >=        | <=
        | less-than | more-than | exactly | at-least | at-most
        |                                    from      | to
```

```
<units> = <unpaired-description>

<num>   = <bounded precision floating point number in LISP format>
```

As described above, intervals can come in one of two forms, the first we call a numeric interval and it describes a range of rational numbers specifying whether or not the endpoints are included. The second form of an interval is simply a numeric interval with a units designation and we represent that as an ordered pair with the numeric interval as the first entry, and with an unpaired description as the second. For example, ((**at-least** 1 **less-than** 10),**liters**) refers to the interval which is the intersection of the numbers greater than or equal to 1, and those strictly less than 10. A single number with a unit specification is also considered to be an numeric interval and hence the intervals (5,**pints**), ((**exactly** 5),**pints**) and ((**at-most** 5 **at-least** 5),**pints**) all mean the same thing.

2.4 Commands

Up to this point we have focused solely on the structure of the language and have not discussed how to change the current knowledge base which the system is acting upon. We now describe the high level commands which change the knowledge base. The basic set is as follows:

```
<command> = (DEF <stmt>)
          | (DCL <stmt>)
          | (TYP <stmt>)
          | (NAME <name> <description>)
```

These commands have the effect of entering statements into the knowledge base. The operators **def**, **dcl**, and **typ** stand for "define", "declare", and "typically" respectively, and essentially they provide the user with a crude qualitative way of identifying the strength of their belief in a given statement. The **name** command allows user to assign a name to given description. One form of command which is clearly missing from this list is query operations which prompt the system to provide the user with information. However, since query statements are somewhat implementation dependent and not conceptually difficult we leave that section of the language unspecified in this thesis.

2.5 Examples

This section gives some more complicated examples of the use of the language to state facts and make definitions:

Ex. 7 *Every patient is treated by some doctor.*
(def ((every **patient**) *treated-by* (some **doctor**)))

If you wanted to say that every patient is treated by the same doctor as every other patient you would have to perform skolemization (e.g. ((every **patient**) *treated-by* Dr-Spock)).

Ex. 8 *Patients are typically treated by a single doctor.*
(typ ((every **patient**) *treated-by* (one **doctor**)))

Ex. 9 *Only doctors prescribe drugs.*

(dcl (every **drug**) (inverse *prescribe*) (only **doctors**))

In these and later examples we allow the plural and singular forms of words to represent the same concept. Hence in this example we use "doctors" in order to increase readability when it should really be the concept, "doctor" that would be in that slot. In the text of the knowledge base used by an implementation, we would either have to use only the singular or only the plural form to refer for example to the description of a doctor, or explicitly state using the name command, that "doctor" and "doctors" really represent the same concept.

Ex. 10 *John's mother's sister is color blind.*
(dcl **John** (chain *mother sister*) (pred **color-blind**))

Ex. 11 *John is a person with a weight of 140 pounds, six feet tall, and who takes an anti-depressant.*

```
(dcl John instance-of
        (restr student (rel weight (num (140,pounds)))
                       (rel height (qty (6,feet)))
                       (rel takes (some anti-depressant))))
```

Ex. 12 *An arch is an artifact with 3 parts: a lintel and two uprights (which are all blocks), and where the uprights support the lintel and there is space between them.*

```
(name arch
    (restr artifact (rel parts (num (3,top-concept))
                    (rel (constrain parts (restr block (pred lintel)))
                         (num (1,top-concept)) (label top))
                    (rel (constrain parts (restr block (pred upright))
                         (num (2,top-concept)) (label support)))
    with (and ((every support) supports (one top))
              ((some support) not-touching (some support)))))
```

Chapter 3

Language Semantics

In this chapter we first introduce the notion of intensional semantics and intensional subsumption by reviewing the traditional semantics given to KL-ONE-like languages and identifying where we depart. Given that we build on the syntax which was outlined in the last chapter and we provide formal semantics for each of those constructs, and an axiomatic inference system for subsumption. We prove that the inference system is sound with respect to our semantics and we conjecture completeness with respect to questions of subsumption.

3.1 Intensional Semantics

Woods [31] proposed replacing the traditional extensional view of concepts and subsumption with intensional interpretations. Unfortunately, Woods did not make these semantical proposals precise, and in this section we present a formal semantic framework intended to capture the distinctions we believe Woods advocated.

3.1.1 Interpretations of Concepts and Roles

Standard treatments of the semantics of KL-ONE-like languages build on the usual notions of logic and model theory, viewing concepts as unary predicates and roles as binary predicates, and interpreting concepts as sets and roles as binary relations over the domain of discourse. More formally, these treatments presume:

- A set $\mathbf{E}$ of entities that constitutes the universe of discourse;
- A set $\mathbf{C}$ of concepts to represent classes of entities, and a disjoint set $\mathbf{R}$ of roles or relations to represent connections between the concepts in $\mathbf{C}$;
- A meaning assignment function $\mu^{\mathbf{e}} : \mathbf{C} \cup \mathbf{R} \rightarrow 2^{\mathbf{E}} \cup 2^{\mathbf{E}\times\mathbf{E}}$.

Normally we think of the function $\mu^{\mathbf{e}}$ as being the union of two more specific functions, $\mu^{\mathbf{e}}_{C} : \mathbf{C} \rightarrow 2^{\mathbf{E}}$ which translates a concept into the set of entities in $\mathbf{E}$ of which it is true, and $\mu^{\mathbf{e}}_{R} : \mathbf{R} \rightarrow 2^{\mathbf{E}\times\mathbf{E}}$ which translates a role into the pairs of entities in $\mathbf{E}$ which satisfy the role. We will call $\mu^{\mathbf{e}}$ an *extensional* meaning function because it maps every concept and role to its denotation.

3.1.2 Subsumption

We say a concept c_1 *subsumes* another concept c_2 with respect to $\mu^{\mathbf{e}}$ if and only if the extension of c_1 includes the extension of c_2 in $\mu^{\mathbf{e}}$, that is iff $c_2) \subseteq \mu^{\mathbf{e}}_{C}(c_1)$. We write this relation as $\mu^{\mathbf{e}} \models (c_1 \textit{ subsumes } c_2)$. We say that c_1 subsumes c_2 without qualification iff c_1 subsumes c_2 with respect to every $\mu^{\mathbf{e}}$, and we write this as $\models (c_1 \textit{ subsumes } c_2)$. We can thus view $(c_1 \textit{ subsumes } c_2)$ as being an abbreviation for $\forall x, c_1(x) \rightarrow c_2(x)$. Thus the concept of a mammal, under the ordinary interpretation, would subsume the concept of a primate which in turn would subsume the concept of a human since the set of human beings is a subset of the set of primates, which is certainly a subset of the set of mammals. The corresponding subsumption relationship for roles is written $\mu^{\mathbf{e}} \models (r_1 \textit{ subsumes } r_2)$, which is true iff $\mu^{\mathbf{e}}_{R}(r_2) \subseteq \mu^{\mathbf{e}}_{R}(r_1)$. As above we can view $(r_1 \textit{ subsumes } r_2)$ as an abbreviation for $\forall x, y, r_1(x, y) \rightarrow r_2(x, y)$. For example the relation brother would be subsumed by the relation sibling which in turn would be subsumed by the relation relative because any two individuals who are brothers must be siblings and thus also relatives. Combining the two notions ideas gives us a form of subsumption that we will refer to as *extensional subsumption* since it is determined completely by the extensions of the various concepts and roles.

Although extensional subsumption has its advantages we are interested in other notions of subsumption for the following three reasons. First is that since extensional subsumption is equivalent to logical implication, it is undecidable for even moderately expressive languages. For example a system seeking to classify concepts by using extensional subsumption would have to determine whether P=NP, if those two complexity classes could even be represented. The second problem is that even when a subsumption question is decidable, it is not necessarily decidable within a reasonable amount of time. Lastly, implication is too broad in the sense that we often want the system to distinguish concepts even when they have the same extension, e.g. humans and featherless bipeds.

3.1.3 A Subsumption Spectrum

In an effort to clarify the alternatives we define several different notions of subsumption, and these views of subsumption constitute a somewhat revised picture of Woods' notions of subsumption as discussed in the previous chapter. The basic idea presented by Woods is to replace a semantic definition of subsumption with an axiomatic one, with subsumption judgments made by checking deriviability of a statement about subsumption.

In order to make this notion more formal we need to presume the existence of a representation language **L**. This language would consist of all the expressions of the kind detailed in the previous chapter for constructing concepts/descriptions and stating their interrelations, and it would also have to be able to make statements of subsumption which would be either true or false given a particular meaning function and a notion of subsumption. Obvious candidates for L include languages such as first order logic (using implication to state subsumption) or that used for KL-ONE.

Given a language **L** we view a knowledge base, KB, as consisting of a collection of sentences in **L**. We say that a given meaning function μ^e satisfies a KB with respect to subsumption if for any statement of the form (c_1 *subsumes* c_2) $\in$ KB, the meaning function entails that relationship, i.e. $\mu^e \models$ (c_1 *subsumes* c_2). In addition we say that a knowledge base entails a given subsumption relationship, i.e. KB $\models$ (c_1 *subsumes*

c_2), if and only if all meaning functions which satisfy KB also entail the relationship (c_1 *subsumes* c_2).

We now detail the following notions of subsumption:

Recorded Subsumption: In this version of subsumption, a concept is considered to subsume another concept if and only if that relation is explicitly stated. Thus KB $\models$ (c_1 *subsumes* c_2) iff (c_1 *subsumes* c_2) $\in$ KB.

Transitive Subsumption: This notion differs from the above only in that it involves the transitive closure of explicitly stated subsumption relations. Hence KB $\models$ (c *subsumes* c') if and only if there exist concepts $c_1, \ldots, c_n$ such that $c_1 = c, c_n = c'$, and (c_i *subsumes* c_{i+1}) $\in$ KB for all i, $1 \leq i \leq n$.

Deductive Subsumption: Deductive subsumption defines an inference relation $\vdash$ on the language L which is sound with respect to $\models$. This inference relation is built upon a set of inference axioms such as, for example, $\forall c_1, c_2\ p(c_1, c_2) \rightarrow$ (c_1 *subsumes* c_2). Thus in this version of subsumption one concept c_1 subsumes another concept c_2 if and only if KB $\vdash$ (c_1 *subsumes* c_2).

For most non-trivial representation languages and knowledge bases the above notions of subsumption span the range of computational difficulty, with recorded subsumption being trivial and deductive subsumption, like extensional subsumption, being intractable. If the relation $\vdash$ is complete with respect to $\models$ then deductive subsumption is equivalent to extensional subsumption, and most KL-ONE languages seem to implement extensional subsumption by means of deductive subsumption. Further, depending on our selection of inference relation, recorded and transitive subsumption can also be viewed as special cases of deductive subsumption.

The focus of our effort is to find some form of deductive subsumption which locates a piece of middle ground between the boundaries of triviality and undecidability. We could simply allow the $\vdash$ relation to be incomplete with respect to $\models$, but effective use of a KR system (or any system) ordinarily requires a relatively clear notion of what inferences the system can and can not draw, and formally characterizing the nature of incomplete deducibility relations is in general a very difficult problem. We instead seek to avoid this difficulty by defining another sense of subsumption, called *intensional* subsumption, which specifies a non-traditional entailment relation $\models$ for which the corresponding sound and complete inference relation $\vdash$ is more easily computable.

3.1.4 Intensional Subsumption

The aim of intensional subsumption is that one concept should intensionally subsume another concept if and only if it is "obvious" from the structure or syntax of the concept that one is more specific than the other. For example one wants to be able to infer purely from the syntactic structure of the concepts c_1 and (and c_1 c_2) and some basic notion of the keyword and, that the concept c_1 is more general than, and thus subsumes the concept (and c_1 c_2). In contrast, the concept of a polygon with three angles and the concept of a polygon with three sides should not be determined to subsume one and other since it is not obvious from the structure of the concepts alone but requires some knowledge about the relationship between the number of sides and the number of angles in a given polygon. Hence the task of our semantics is to provide a reasonable formal distinction between these extremes.

The central theoretical idea which undergirds the semantics of intensional subsumption is a shift in the universe of discourse away from arbitrary entities to a simpler domain, the set of *conceptual descriptions*, which we take to consist of the linguistic constructs of the systems: concepts, roles, etc. We view conceptual descriptions as stand-ins for their referents. Statements about descriptions are taken to be statements about the descriptions themselves rather than their referents, in contrast with the usual view of statements in most KL-ONE-like languages. In such systems the only meaningful statements involving concepts are statements about the concepts' referents since concepts are equated with logical predicates which are interpreted as sets, and the only statements which can be made about a set are about its membership. This therefore makes our task easier in the sense of ensuring soundness and completeness since we do not have to worry about unintended or non-existent referents.

The formal semantics are fairly straightforward. We first define the set of descriptions $\mathbf{D}$ which is a superset of the set of concepts, relations, and others, and then we define an intensional meaning function $\mu^{\mathbf{i}} : \mathbf{D} \rightarrow \mathcal{V}(\mathbf{D})$ which maps a given description d into the universe of sets over the elements of $\mathbf{D}$. Given a description d, the expression $\mu^{\mathbf{i}}(d)$ defines a set such that every element of the set can be viewed as *satisfying* the description d since it either describes an instance or a subdescription.

In addition to the function $\mu^{\mathbf{i}}$, we define two additional meaning functions which are specializations of the function $\mu^{\mathbf{i}}$, namely $\mu^{\mathbf{i}}_C : \mathbf{D} \rightarrow 2^{\mathbf{D}}$ and $\mu^{\mathbf{i}}_R : \mathbf{D} \rightarrow 2^{\mathbf{D}\times\mathbf{D}}$ where for every d, $\mu^{\mathbf{i}}_C(d) = (\mathbf{D} \cap \mu^{\mathbf{i}}(d))$ and $\mu^{\mathbf{i}}_R(d) = ((\mathbf{D} \times \mathbf{D}) \cap \mu^{\mathbf{i}}(d))$. The function $\mu^{\mathbf{i}}_C$ maps a description to a set of descriptions such that each description satisfies the description d (i.e. the concept view of a description), and the function $\mu^{\mathbf{i}}_R$ maps a description onto a set of ordered pairs of descriptions representing the set of pairs of descriptions for which the described relationship holds. These functions correspond to the functions $\mu^{\mathbf{e}}_C$ and $\mu^{\mathbf{e}}_R$ from the KL-ONE discussion, but where the KL-ONE functions can only be meaningfully applied to some elements of the model domain, our functions can be applied to all descriptions. It is worth noting that this system may be overly general in its use of term $\mathcal{V}(\mathbf{D})$ as the range for our meaning function, however it leaves room for expanding the notion of descriptions to include other notions, including for example any arbitrary function.

3.2 Semantics of Descriptions

We first present the semantics of descriptions, then of commands, and finally the axioms of the subsumption procedure. The notation of the following discussion assumes that c represents a concept, m a generic modifier, d an element of $\mathcal{D}$, and r a two-place relation of the language. When the meaning function $\mu^{\mathbf{i}}$ is understood the expression $d_1 \mapsto d_2$ abbreviates $d_1 \in \mu^{\mathbf{i}}(d_2)$. Furthermore, in order to avoid confusion with other variable we will use μ to abbreviate $\mu^{\mathbf{i}}$ for remained of the chapter.

3.2.1 Concept

The semantics for the built-in concepts are as follows:

- $\mu(\textbf{top-concept}) = \mathcal{D}$
- $\mu(\textbf{true}) = \mathcal{D}$

- $\mu(\mathbf{false}) = \emptyset$

There are four constructs we can use to build complex concepts:

- $\mu((\textsf{and}\ c_1\ c_2 \ldots c_n)) = \mu(c_1) \cap \mu(c_2) \cap \ldots \mu(c_n)$
- $\mu((\textsf{or}\ c_1\ c_2 \ldots c_n)) = \mu(c_1) \cup \mu(c_2) \cup \ldots \mu(c_n)$
- $\mu((\textsf{not}\ c)) = \{d \in \mathcal{D} \mid d \notin \mu(c)\}$

The last operator for constructing concept is the **restr** which we will discuss in a section of its own shortly.

3.2.2 Intervals

One of the principle functions of intervals is as elements of statements and modifiers as will be described shortly, however intervals are also descriptions in their own right and can be satisfied by other descriptions. Since we represent numbers using LISP floats let us define the set $\mathbf{F}$ of such numbers. Thus given a numeric interval i, the expression $\mu(i)$ is equal to the set of elements of $\mathbf{F}$ which lie within the interval described by i. So for example $\mu((\textsf{at-least}\ x\ \textsf{less-than}\ y)) = \{d \in \mathbf{F} \mid (d > x) \wedge (d \leq y)\}$. The meaning of numeric intervals using other `numeric-rel-op`'s are all very straightforward. The meaning of intervals with unit specification is is a special case of the meaning of paired descriptions which is as follows:

- $\mu((d_1, d_2)) = \{(d_3, d_4) \mid (d_3 \in \mu(d_1)) \wedge (d_4 \in \mu(d_2))\}$

3.2.3 Built-in Relations

This section presents the semantics of the built-in relations. The relations that can be used between any two descriptions are:

- $(d_1, d_2) \in \mu(\textit{subsumes})$ iff $\mu(d_1) \subseteq \mu(d_2)$.
- $(d_1, d_2) \in \mu(\textit{instance-of})$ iff $d_1 \in \mu(d_2)$.

- $(d_1, d_2) \in \mu(same\text{-}as)$ iff $\mu(d_1) = \mu(d_2)$.

The following relations are intended to be used only between descriptions of certain types:

- $(m_1, m_2) \in \mu(specializes)$ iff $\mu(m_1) \subseteq \mu(m_2)$.
- $(c_1, c_2) \in \mu(kind\text{-}of)$ iff $\mu(c_1) \subseteq \mu(c_2)$.
- $(d, m) \in \mu(satisfies)$ iff $d \in \mu(m)$.

Although the semantics of the this set of relations are identical to the semantics of some of the semantics of relations in the preceding set, we create and encourage the use of these two different sets of relations in order to preserve the distinction between modifiers and concepts discussed in section 2.1.2.

3.2.4 Relation-forming Operators

The following describes the semantics of the relation forming operators:

- $(d_1, d_2) \in \mu((\mathsf{inverse}\ r))$ iff $(d_2, d_1) \in \mu(r)$.
- $(d_1, d_2) \in \mu((\mathsf{all}\ r))$ iff $\{d \in \mathcal{D} \mid (d_1, d) \in \mu(r)\} = \mu(d_2)$.
- $(d_0, d_n) \in \mu((\mathsf{chain}\ r_1\ r_2\ \ldots r_n))$ iff $n \geq 2$ and $\exists d_1, d_2, \ldots, d_{n-1} \in \mathcal{D}, (d_0\ r_1\ d_1) \wedge (d_1\ r_2\ d_2) \wedge \ldots (d_{n-2}\ r_{n-1}\ d_{n-1}) \wedge (d_{n-1}\ r_n\ d_n)$.
- $(d_1, d_3) \in \mu((\mathsf{constrain}\ r\ d_2))$ iff $((d_1, d_3) \in r) \wedge (d_3 \in \mu(d_2))$.

The **all** relation says that the second description describes the set of all descriptions which stand in the specified relationship to the first description. The **chain** relation allows one to assert that two descriptions are connected by a chain of relations and intervening descriptions, and the **constrain** relations restricts the range of a relation to only those concepts which satisfy a given description.

3.2.5 Statements

Before we can present the semantics for several key elements of the language we need to discuss the semantics of statements. For any given statement it must be satisfied by only one of the built-in concepts **true** or **false**. We say that a statement s is true (w.r.t. μ) if $\mu(s) = \{\textbf{true}\}$ and s is false (w.r.t. μ) if $\mu(s) = \{\textbf{false}\}$. The simplest form of a statement is $(d_1\ r\ d_2)$, and in that case $\mu((d_1\ r\ d_2)) = \{\textbf{true}\}$ iff $(d_1, d_2) \in \mu(r)$, otherwise $\mu((d_1\ r\ d_2)) = \{\textbf{false}\}$. We will discuss shortly the semantics of more complex statements involving quantified descriptions, but now we present the semantics of the basic compound statements:

- $\mu((\mathsf{or}\ s_1\ s_2 \ldots s_n) = \{\textbf{true}\}$ iff for some i where $1 \geq i \geq n$, $\mu(s_i) = \{\textbf{true}\}$.
- $\mu((\mathsf{and}\ s_1\ s_2 \ldots s_n) = \{\textbf{true}\}$ iff for all i where $1 \geq i \geq n$, $\mu(s_i) = \{\textbf{true}\}$.
- $\mu((\mathsf{not}\ s)) = \{\textbf{true}\}$ iff $\mu(s) = \{\textbf{false}\}$.

3.2.6 Quantified Descriptions

In this section we will describe how the above semantics are changed by the introduction of quantificational information. To do this we will need to introduce some additional functions to explain the semantics; the basic format is as follows:

- $\mu(((Tag_1\ d_1)\ r\ (Tag_2\ d_2)))$=$\{\textbf{true}\}$ iff $\varphi_1(Tag_1, d_1, r, Tag_2, d_2)$.
- $\mu((d_1\ r\ (Tag_2\ d_2)))$=$\{\textbf{true}\}$ iff $\varphi_1(\text{"No-Tag"}, d_1, r, Tag_2, d_2)$.
- $\mu(((Tag_1\ d_1)\ r\ d_2))$=$\{\textbf{true}\}$ iff $\varphi_1(Tag_1, d_1, r, \text{"No-Tag"}, d_2)$.

The function φ_1 and an auxiliary function φ_2 are defined in the Table 3.1. Note that in the last two entries for φ_2 the tag is numeric so we require d_2 to be an interval.

3.2.7 Examples

The following provides some examples of uses of the above formulas to generate the semantics of simple statements:

Expression	*Meaning*
$\varphi_1(\text{“No-Tag”}, d_1, r, Tag_2, d_2)$	$\varphi_2(d_1, r, Tag_2, d_2)$
$\varphi_1(\text{Some}, d_1, r, Tag_2, d_2)$	$d \in \mathcal{D}, (d \mapsto d_1) \land \varphi_2(d, r, Tag_2, d_2)$
$\varphi_1(\text{Every}, d_1, r, Tag_2, d_2)$	$\forall d \in \mathcal{D}, (d \mapsto d_1) \to \varphi_2(d, r, Tag_2, d_2)$
$\varphi_1(\text{Only}, d_1, r, Tag_2, d_2)$	$\forall d \in \mathcal{D}, \varphi_2(d, r, Tag_2, d_2) \to (d \mapsto d_1)$
$\varphi_1(\text{No}, d_1, r, Tag_2, d_2)$	$\forall d \in \mathcal{D}, \varphi_2(d, r, Tag_2, d_2) \to \neg(d \mapsto d_1)$
$\varphi_1(\text{One}, d_1, r, Tag_2, d_2)$	$\exists d \in \mathcal{D}, (d \mapsto d_1) \land \varphi_2(d, r, Tag_2, d_2) \land (\forall d^* \in \mathcal{D}, ((d^* \mapsto d_1) \land \varphi_2(d^*, r, Tag_2, d_2)) \leftrightarrow (d^* = d))$

Expression	*Meaning*
$\varphi_2(d, r, \text{“No-Tag”}, d_2)$	$(d\ r\ d_2)$
$\varphi_2(d, r, \text{Some}, d_2)$	$\left(\exists \bar{d} \in \mathcal{D}, (\bar{d} \mapsto d_2) \land ((d, \bar{d}) \in \mu(r))\right)$
$\varphi_2(d, r, \text{One}, d_2)$	$\left(\exists \bar{d} \in \mathcal{D}, (\bar{d} \mapsto d_2) \land ((d, \bar{d}) \in \mu(r)) \land (\forall d' \in \mathcal{D}, ((d' \mapsto d_2) \land ((d, d') \in \mu(r))) \leftrightarrow (d' = \bar{d})\right)$
$\varphi_2(d, r, \text{Every}, d_2)$	$\left(\forall \bar{d} \in \mathcal{D}, (\bar{d} \mapsto d_2) \to ((d, \bar{d}) \in \mu(r))\right)$
$\varphi_2(d, r, \text{Only}, d_2)$	$\left(\forall \bar{d} \in \mathcal{D}, ((d, \bar{d}) \in \mu(r)) \to (\bar{d} \mapsto d_2)\right)$
$\varphi_2(d, r, \text{No}, d_2)$	$\left(\forall \bar{d} \in \mathcal{D}, ((d, \bar{d}) \in \mu(r)) \to \neg(\bar{d} \mapsto d_2)\right)$
$\varphi_2(d, r, \text{Qty}, d_2)$	$\exists d \in \mathcal{D}, (d \mapsto d_2) \land ((d, d) \in \mu(r))$
$\varphi_2(d, r, \text{Num}, d_2)$	$(\exists d_3, d_4 \in \mathcal{D} \quad d_2 = (d_3, d_4)) \land \lvert \{\bar{d} \in \mathcal{D} \mid (\bar{d} \mapsto d_4) \land (d, \bar{d}) \mapsto r\} \rvert \in \mu(d_3)$

Table 3.1: Semantics of statements involving quantified descriptions.

Ex. 13 ((Every **Ovary**) *contains* (Some **Egg**)) is true iff:

$\forall d \in \mathcal{D}, (d \mapsto \textbf{Ovary}) \rightarrow \left(\exists \bar{d}, (\bar{d} \mapsto \textbf{Egg}) \wedge ((d, \bar{d}) \in \mu(\textit{contains}))\right)$

Ex. 14

((Every **Fetus**) *fed-through* (One **Placenta**)) is true iff:

$$\forall d \in \mathcal{D}, (d \mapsto \textbf{Fetus}) \rightarrow \big(\exists \bar{d} \in \mathcal{D}, (\bar{d} \mapsto \textbf{Placenta}) \wedge (d \textit{ fed-through } \bar{d}) \wedge$$
$$(\forall d' \in \mathcal{D}, ((d' \mapsto \textbf{Placenta}) \wedge (d' \neq \bar{d})) \rightarrow$$
$$\neg(d \textit{ fed-through } d'))$$

Ex. 15

((No **Contraceptive**) *prevents* (Every **STD**)) is true iff:

$$\forall d \in \mathcal{D}, ((\forall \bar{d} \in \mathcal{D}, (\bar{d} \mapsto \textbf{STD}) \rightarrow (d \textit{ prevents } \bar{d})) \wedge (\exists \bar{d} \in \mathcal{D}, (\bar{d} \mapsto \textbf{STD})))$$
$$\rightarrow \neg(d \mapsto \textbf{Contraceptive})$$

The concept **STD** abbreviates **Sexually-Transmitted-Disease**.

Ex. 16 (**Sally** *eats* (Only **Kosher-Foods**)) is true iff:

$\forall \bar{d} \in \mathcal{D}, (\textbf{Sally} \textit{ eats } \bar{d}) \rightarrow (\bar{d} \mapsto \textbf{Kosher-Foods})$

Ex. 17

((No **Contraceptive**) *prevents* (Only **STDs**)) is true iff:

$$\forall d \in \mathcal{D}, ((\forall \bar{d} \in \mathcal{D}, (d \textit{ prevents } \bar{d}) \rightarrow (\bar{d} \mapsto \textbf{STD})) \wedge (\exists \bar{d} \in \mathcal{D}, (d \textit{ prevents } \bar{d})))$$
$$\rightarrow \neg(d \mapsto \textbf{Contraceptive})$$

Ex. 18

(**Sally** (Constrain *uses* **Contraceptives**) (Only **Oral-Contraceptives**)) is true iff:

$$\forall \bar{d} \in \mathcal{D}, ((\textbf{Sally} \textit{ uses } \bar{d}) \wedge (\bar{d} \mapsto \textbf{Contraceptive})) \rightarrow$$
$$(\bar{d} \mapsto \textbf{Oral-Contraceptive})$$

Ex. 19 ((Only **Men**) *use* **Condoms**) is true iff:

$\forall d \in \mathcal{D}, (d \textit{ uses } \textbf{Condoms}) \rightarrow (d \mapsto \textbf{Man})$

Ex. 20

(**Sally** (All (Constrain *sexual-partner* **Man**)) (restr **Man** (rel *co-worker* **Sally**))) is true iff:

$$\forall d \in \mathcal{D},((\textbf{Sally}, d) \mapsto \textit{sexual-partners}) \wedge (d \mapsto \textbf{Man})) \leftrightarrow$$
$$((d \mapsto \textbf{Man}) \wedge ((d, \textbf{Sally}) \mapsto \textit{co-worker})))$$

3.2.8 Modifiers

The construction of modifiers relies on the construction of quantified descriptions as described in the previous sections. Recall from the previous chapter that modifiers are intended to represent the traditional notion of a predicate and modifiers without label statements can take one of following forms:

- $\mu((\mathsf{rel}\ r\ d)) = \{\hat{d} \mid (\hat{d}, d) \in \mu(r)\}$.
- $\mu((\mathsf{rel}\ r\ (\textit{Tag}\ d))) = \{d_1 \mid \varphi_2(d_1, r, \textit{Tag}, d)\}$.
- $\mu((\mathsf{pred}\ d)) = \mu(d)$.

The value of φ_2 is described in table 3.1. Since the use of label statements within modifiers is only meaningful when labels appear within a restr, we discuss that case in conjunction with the restr operator, in the next section.

3.2.9 Restriction Concepts

Recall that the restr operator returns a concept. First we present the form which does not use labels or statements which is as follows:

- $$\begin{aligned}\mu((\mathsf{restr}\ c\ &(\mathsf{rel}\ r_1\ (\textit{Tag}_1\ \hat{d}_1))\\ &(\mathsf{rel}\ r_2\ (\textit{Tag}_2\ \hat{d}_2))\\ &\vdots\\ &(\mathsf{rel}\ r_n\ (\textit{Tag}_n\ \hat{d}_3))) = \{d \in \mathcal{D} \mid (d \in \mu(c)) \wedge \varphi_2(d, r_1, \textit{Tag}, \hat{d}_1)\\ &\qquad \wedge \varphi_2(d, r_2, \textit{Tag}, \hat{d}_2)\\ &\qquad \vdots\\ &\qquad \wedge \varphi_2(d, r_n, \textit{Tag}, \hat{d}_n)\}\end{aligned}$$

If every modifier is labelled and there is a statement attached to the end of the restr operator then the semantics are listed below. In the cases in which only some modifiers have labels or tags, the semantics can be easily determined, however describing those cases adds still more complexity so we omit them at this point.

- $\mu((\textsf{restr}\ c\ (\textsf{rel}\ r_1\ (Tag_1\ \hat{d}_1)\ (\textsf{label}\ l_1))$
  $(\textsf{rel}\ r_2\ (Tag_2\ \hat{d}_2)\ (\textsf{label}\ l_2))$
  $\vdots$
  $(\textsf{rel}\ r_n\ (Tag_n\ \hat{d}_3)\ (\textsf{label}\ l_n))$
  $\textsf{with}\ ((Tag\ d_1)\ r\ (Tag'\ d_2))))=$

$$\{d \in \mathcal{D} \mid (d \in \mu(c)) \wedge \varphi_2(d, r_1, Tag, \hat{d}_1) \wedge \varphi_2(d, r_2, Tag, \hat{d}_2) \ldots \wedge \varphi_2(d, r_n, Tag, \hat{d}_n) \wedge$$
$$\varphi_1(Tag, d_1, r, Tag', d_2)[\mu(l_1)/\{\hat{d} \mid (d, \hat{d}) \in \mu(r_1)\},$$
$$\mu(l_2)/\{\hat{d} \mid (d, \hat{d}) \in \mu(r_2)\},$$
$$\vdots$$
$$\mu(l_n)/\{\hat{d} \mid (d, \hat{d}) \in \mu(r_n)\}]\}$$

Although the expression above is complex consider the following example from Chapter 2:

(restr **student** (rel *favorite-professor* (one **Person**) (label **L1**))
(rel *taking* (some **class**) (label **L2**))
with ((one **L1**) *teaches* (some **L2**)))

In this case we are describing a student who has exactly one favorite professor and that professor is teaching some class that the student is taking. The crucial element of the semantics above is the substitution of a new meaning for l_i and that this meaning depends on d. Thus the concepts **L1** and **L2** do not have a static meaning such that it is possible to describe $\mu(\mathbf{L1})$ or $\mu(\mathbf{L2})$. In a sense we are using the labels to perform a syntactic trick to get the desired semantics. Finally, note that although the tag in front of **L2** must be some, the tag in front of **L1** could be either one, some, or every since we can assume that the cardinality of $\mu(\mathbf{L2})$ is always one.

3.3 Semantics of Commands

In the current version of the language there are only two types of commands. Commands prefixed with either **def**, **dcl**, or **dcl** we call *assertions*. The effect of assertions is simply to add a statement to the knowledge base, and we know that for any statement s, if $s \in$KB then $\mu(s) = \{\textbf{true}\}$. Assertions also mark the statement as either definitional, assertional, or defeasible. Unfortunately we have not developed formal notions of how these markings would effect the semantics and thus for now they are ignored on that level. We believe, however that allowing the user to preserve this distinction on some level is of high importance. In place of formal notions we will mention briefly two proposals for directions along which formal semantics might be developed, although this certainly does not do justice to the broad body of literature which addresses the integration of definitional, assertional, and defeasible information.

One option builds on the notion of epistemic entrenchment by suggesting that definitional information is the very last kind of information to relax in the face of a contradiction and brings into question the very nature of entity involved, whereas assertional information is something that is know to be true of a given object or category of objects but which is not central to the nature of the entity. For example, if we made the assertion that infants do not play chess that statement would be assertional, since if one day there was an infant born of a grand-master who could play a brief game of chess in her third day of life that would be extraordinary but it would not call into question whether the child was an "infant" or not. Whereas, if woman gave birth to a fully mature adult midget we might call such a being an "infant".

Another possible proposal is to view the difference between definitional, assertional, and typical information using the notion of orders of magnitude in probability. By that scheme we would view definitional information as facts which cannot be contradicted without forcing the relevant concept to lose most of its intended meaning (i.e. probability of a contradiction is 0), assertional information represents material which we believe with near certainty (the sun will rise tomorrow) but for which we can conceive of a scenario in which it would not be true (the earth explodes) (i.e. possibility of a contradiction is very small but non-zero). Finally, typicality assertions are ones which we can easily conceive of either being true or false (i.e. probability order of 10^{-1}) but which are more likely to be true than not true.

The last command which we need to discuss is the **name** command, which at first glance seems to be very similar to the **dcl**, **def**, or **typ** commands. However the **name** command actually has a very different use, essentially it acts as a macro by creating a pointer to a given concept. Thus where (**dcl triangle** *same-as* (**restr** ...)) would cause the system to create two new description records or structures and assert the *same-as* relation between them, the command **name triangle** (**restr** ...)) creates only one new structure which can be referenced through the name "triangle".

3.4 Rules of Subsumption

At the beginning of this chapter we introduced the notion of an inference relation and in this section we detail the rules which form the basis of our particular inference relation $\vdash_i$. The format of this section will be that we will state each rule and then immediately prove it to be sound with respect to the semantics from the preceding sections. Most of the rules are straightforward although there are a few in which generalizing a subexpression of a description produces a more specific description. One piece of notation we will need is of the form $\vdash_i$ (d_1 *subsumes* d_2), which indicates that for all knowledge bases, KB $\vdash_i$ (d_1 *subsumes* d_2). In the remainder of this discussion we will also drop the subscript from $\vdash_i$ to avoid confusion.

3.4.1 Built-in Relations and Base Axioms

The following are the most basic rules involving the built-in subsumption relations and the identity and transitivity rules. Note that our first rules requires no hypothesis.

- $\overline{(d \textit{ subsumes } d)}$
  Soundness Proof: $\mu(d) \subseteq \mu(d)$ so (d *subsumes* d).

- $\frac{(d_1 \textit{ subsumes } d_3),(d_3 \textit{ subsumes } d_2)}{(d_1 \textit{ subsumes } d_2)}$

  Soundness Proof: We are given that $\mu(d_2) \subseteq \mu(d_3) \subseteq \mu(d_1)$, so by the transitivity of the subset relation $\mu(d_2) \subseteq \mu(d_1)$. Thus $(d_1 \textit{ subsumes } d_2)$.

- $\frac{(m_2 \textit{ specializes } m_1)}{(m_1 \textit{ subsumes } m_2)}$

  Soundness Proof: By definition.

- $\frac{(c_2 \textit{ kind-of } c_1)}{(c_1 \textit{ subsumes } c_2)}$

  Soundness Proof: By definition.

- $\frac{(d_2 \textit{ same-as } d_1)}{(d_1 \textit{ subsumes } d_2)}$

  Soundness Proof: By definition.

- $\frac{(d_1 \textit{ same-as } d_2)}{(d_1 \textit{ subsumes } d_2)}$

  Soundness Proof: By definition.

- $\frac{((\mathsf{Every}\ d_1) \textit{ instance-of } c_1)}{(c_1 \textit{ subsumes } d_1)}$

  Soundness Proof: The meaning of $((\mathsf{Every}\ d_1) \textit{ instance-of } c_1)$ following the tables describing the semantics of quantification tags in section 3.2.6 is $\forall d \in \mathcal{D}, (d \mapsto d_1) \rightarrow (d, c_1) \in \mu(\textit{instance-of})$. Using the semantics of the *instance-of* relation from section 3.2.3 the above implies that $\forall d \in \mathcal{D}, d \in \mu(d_1) \rightarrow d \in \mu(c_1)$. Thus $\mu(d_1) \subseteq \mu(c_1)$.

- $\frac{((\mathsf{Every}\ d_1) \textit{ satisfies } m_1)}{(m_1 \textit{ subsumes } d_1)}$

  Soundness Proof: Following the format of the above proof, the meaning of $((\mathsf{Every}\ d_1) \textit{ satisfies } m_1)$ from the tables is $\forall d \in \mathcal{D}, (d \mapsto d_1) \rightarrow (d, m_1) \in \mu(\textit{satisfies}))$. Substituting the meaning of the *satisfies* relation implies that $\forall d \in \mathcal{D}, d \in \mu(d_1) \rightarrow d \in \mu(m_1)$. Thus $\mu(d_1) \subseteq \mu(m_1)$.

3.4.2 Concepts

- $\frac{}{(\textbf{top-concept} \textit{ subsumes } c)}$

  Soundness Proof: By definition.

- $$\frac{(c_1 \textit{ subsumes } c'_1), (c_2 \textit{ subsumes } c'_2), \ldots (c_n \textit{ subsumes } c'_n)}{(\textsf{and } c_1\ c_2 \ldots c_n) \textit{ subsumes } (\textsf{and } \hat{c}_1\ \hat{c}_2 \ldots \hat{c}_m)}$$
where for all i where $0 \leq i \leq n$ there exists some j where $0 \leq j \leq m$ such that $c'_i = \hat{c}_j$

  Soundness Proof: If $d \in \mu((\textsf{and } \hat{c}_1\ \hat{c}_2 \ldots \hat{c}_m))$ then by definition $d \in \mu(\hat{c}_j)$ for all j, $1 \leq j \leq m$. However, for each c_i there is some $\hat{c}_j$ such that (c_i *subsumes* $\hat{c}_j$), i.e. $\mu(\hat{c}_j) \subseteq \mu(c_i)$. So since $d \in \mu(\hat{c}_j)$ for all j it follows that $d \in \mu(c_i)$ for all i and therefore $d \in \mu((\textsf{and } c_1\ c_2 \ldots c_n))$, hence $\mu((\textsf{and } c_1\ c_2 \ldots c_n)) \subseteq \mu((\textsf{and } \hat{c}_1\ \hat{c}_2 \ldots \hat{c}_m))$.

- $$\frac{(c'_1 \textit{ subsumes } \hat{c}_1), (c'_2 \textit{ subsumes } \hat{c}_2), \ldots (c'_n \textit{ subsumes } \hat{c}_m)}{(\textsf{or } c_1\ c_2 \ldots c_n) \textit{ subsumes } (\textsf{or } \hat{c}_1\ \hat{c}_2 \ldots \hat{c}_m)}$$
where for all j where $0 \leq j \leq m$ there exists some i where $0 \leq i \leq n$ such that $c'_j = c_i$.

  Soundness Proof: If $d \in \mu((\textsf{or } \hat{c}_1\ \hat{c}_2 \ldots \hat{c}_m))$ then by definition there exists some j' such that $d \in \mu(\hat{c}_{j'})$. From the givens it follows that there exists some i' such that ($c_{i'}$ *subsumes* $\hat{c}_{j'}$) and thus $d \in \mu(c_{i'})$. Therefore $d \in \mu((\textsf{or } c_1\ c_2 \ldots c_n)$, so $\mu((\textsf{or } \hat{c}_1\ \hat{c}_2 \ldots \hat{c}_m)) \subseteq \mu((\textsf{or } c_1\ c_2 \ldots c_n))$.

- $$\frac{(c_2 \textit{ subsumes } c_1)}{((\textsf{not } c_1) \textit{ subsumes } (\textsf{not } c_2))}$$
Soundness Proof: If $d \in \mu((\textsf{not } c_2))$ then $d \notin \mu(c_2)$. So if $\mu(c_1) \subseteq \mu(c_2)$ then we know that $d \notin \mu(c_1)$ and thus $d \in \mu((\textsf{not } c_1))$ so $\mu((\textsf{not } c_2)) \subseteq \mu((\textsf{not } c_1))$.

- $$\frac{(c_1 \textit{ subsumes } c_2), (m_1 \textit{ subsumes } m'_1), (m_2 \textit{ subsumes } m'_2), \ldots (m_n \textit{ subsumes } m'_n)}{((\textsf{restr } c_1\ m_1\ m_2 \ldots m_n) \textit{ subsumes } (\textsf{restr } c_2\ \hat{m}_1\ \hat{m}_2 \ldots \hat{m}_m))}$$
where for all i where $0 \leq i \leq n$ there exists some j where $0 \leq j \leq m$ such that $m'_i = \hat{c}_j$.

  Soundness Proof: If $d \in \mu((\textsf{restr } c_2\ \hat{m}_1\ \hat{m}_2 \ldots \hat{m}_m))$ then by definition $d \in \mu(c_2)$ and $d \in \mu(\hat{m}_j)$ for all j, $1 \leq j \leq m$. However, since (c_1 *subsumes* c_2), $d \in \mu(c_1)$ and for each m_i there is some $\hat{m}_j$ such that (m_i *subsumes* $\hat{m}_j$), i.e. $\mu(\hat{m}_j) \subseteq \mu(m_i)$. So since $d \in \mu(\hat{m}_j)$ for all j it follows that $d \in \mu(m_i)$ for all i and therefore $d \in \mu((\textsf{rest } c_1\ m_1\ m_2 \ldots m_n))$, hence $\mu((\textsf{restr } c_1\ m_1\ m_2 \ldots m_n)) \subseteq \mu((\textsf{restr } c_2\ \hat{m}_1\ \hat{m}_2 \ldots \hat{m}_m))$.

3.4.3 Modifiers

- $\frac{(r_2 \textit{ subsumes } r_1),(d_1 \textit{ subsumes } d_2)}{((\mathsf{rel}\ r_1\ (\mathsf{only}\ d_1))\ \textit{subsumes}\ (\mathsf{rel}\ r_2\ (\mathsf{only}\ d_2)))}$

  Soundness Proof: If $d \in \mu((\mathsf{rel}\ r_2\ (\mathsf{only}\ d_2)))$ then $\forall \hat{d} \in \mathcal{D}$ such that $(d, \hat{d}) \in \mu(r_2)$ it follows by definition that $\hat{d} \in \mu(d_2)$. Since $\mu(r_1) \subseteq \mu(r_2)$ we know that $\{\hat{d} \in \mathcal{D} \mid (d, \hat{d}) \in \mu(r_1)\} \subseteq \{\hat{d} \in \mathcal{D} \mid (d, \hat{d}) \in \mu(r_2)\}$. Further since ($d_1$ *subsumes* d_2) it is clear that if $\hat{d} \in \mu(d_2)$ then $\hat{d} \in \mu(d_1)$. If $\forall \hat{d} \in \mathcal{D}$ s.t. $(d, \hat{d}) \in \mu(r_2)$ implies $\hat{d} \in \mu(d_2)$ then $\forall \hat{d} \in \mathcal{D}$ s.t. $(d, \hat{d}) \in \mu(r_1)$ implies $\hat{d} \in \mu(d_1)$ since both the condition and the implication are weaker in the second statement. Thus $d \in \mu((\mathsf{rel}\ r_1\ (\mathsf{only}\ d_1)))$ and hence $\mu((\mathsf{rel}\ r_1\ (\mathsf{only}\ d_1))) \subseteq \mu((\mathsf{rel}\ r_2\ (\mathsf{only}\ d_2)))$.

- $\frac{(d_2 \textit{ subsumes } d_1),(r_1 \textit{ subsumes } r_2)}{((\mathsf{rel}\ r_1\ (\mathsf{every}\ d_1))\ \textit{subsumes}\ (\mathsf{rel}\ r_2\ (\mathsf{every}\ d_2)))}$

  Soundness Proof: If $d \in \mu((\mathsf{rel}\ r_2\ (\mathsf{every}\ d_2)))$ then $\forall \hat{d} \in \mu(d_2)$, $(d, \hat{d}) \in \mu(r_2)$ by definition. Since (d_2 *subsumes* d_1) it follows that $\forall d' \in \mu(d_2)$ implies that $d' \in \mu(d_1)$ and hence $(d, d') \in \mu(r_2)$. Further since (r_1 *subsumes* r_2), it is also true that $(d, d') \in \mu(r_1)$. Thus $\forall d' \in \mu(d_1), (d, d') \in \mu(r_1)$ and therefore $d \in \mu((\mathsf{rel}\ r_1\ (\mathsf{every}\ d_1)))$. So $\mu((\mathsf{rel}\ r_1\ (\mathsf{every}\ d_1))) \subseteq \mu(\mathsf{rel}\ r_2\ (\mathsf{every}\ d_2))$.

- $\frac{(d_1 \textit{ subsumes } d_2),(r_1 \textit{ subsumes } r_2)}{((\mathsf{rel}\ r_1\ (\mathsf{some}\ d_1))\ \textit{subsumes}\ (\mathsf{rel}\ r_2\ (\mathsf{some}\ d_2)))}$

  Soundness Proof: If $d \in \mu((\mathsf{rel}\ r_2\ (\mathsf{some}\ d_2))$ then by definition $\exists \hat{d} \in \mu(d_2)$ s.t. $(d, \hat{d}) \in \mu(r_2)$. It is also true that $\hat{d} \in \mu(d_1)$ since (d_1 *subsumes* d_2), and that $(d, \hat{d}) \in \mu(r_1)$ since (r_1 *subsumes* r_2). Thus $\exists \hat{d} \in \mu(d_1)$ and $(d, \hat{d}) \in \mu(r_1)$ so $d \in \mu((\mathsf{rel}\ r_1\ (\mathsf{some}\ d_1))$ and therefore $\mu((\mathsf{rel}\ r_2\ (\mathsf{some}\ d_2)) \subseteq \mu((\mathsf{rel}\ r_1\ (\mathsf{some}\ d_1))$.

- $\frac{(d_2 \textit{ subsumes } d_1),(r_2 \textit{ subsumes } r_1)}{((\mathsf{rel}\ r_1\ (\mathsf{no}\ d_1))\ \textit{subsumes}\ (\mathsf{rel}\ r_2\ (\mathsf{no}\ d_2)))}$

  Soundness Proof: If $d \in \mu((\mathsf{rel}\ r_2\ (\mathsf{no}\ d_2)))$ then $\forall \hat{d} \in \mu(d_2), (d, \hat{d}) \notin \mu(r_2)$. Since ($d_2$ *subsumes* d_1) and (r_2 *subsumes* r_1) we know that $\forall \hat{d} \in \mu(d_1), (d, \hat{d}) \notin \mu(r_1)$. Thus $d \in \mu((\mathsf{rel}\ r_1\ (\mathsf{no}\ d_1)))$ and therefore $\mu((\mathsf{rel}\ r_1\ (\mathsf{no}\ d_1))) \subseteq \mu((\mathsf{rel}\ r_2\ (\mathsf{no}\ d_2)))$.

- $\frac{(d_1 \textit{ subsumes } d_2),(r_1 \textit{ subsumes } r_2)}{((\mathsf{rel}\ r_1\ (\mathsf{qty}\ d_1))\ \textit{subsumes}\ (\mathsf{rel}\ r_2\ (\mathsf{qty}\ d_2)))}$

  Soundness Proof: If $d \in \mu((\mathsf{rel}\ r_2\ (\mathsf{qty}\ d_2)))$ then $\exists i \in \mathcal{I}$, where $\mathcal{I}$ is the set of interval descriptions, such that $i \in \mu(d_2)$ and $(d,i) \in \mu(r_1)$. Since (d_1 *subsumes* d_2) and (r_1 *subsumes* r_2) it is also true that $i \in \mu(d_1)$ and $(d,i) \in \mu(r_1)$. Thus $\exists i \in \mathcal{I}$ such that $i \in \mu(d_1)$ and $(d,i) \in \mu(r_1)$. Therefore, $d \in \mu((\mathsf{rel}\ r_1\ (\mathsf{qty}\ d_1)))$ and thus $\mu((\mathsf{rel}\ r_2\ (\mathsf{qty}\ d_2))) \subseteq \mu((\mathsf{rel}\ r_1\ (\mathsf{qty}\ d_1))$.

- $$\frac{(r_1 \textit{ subsumes } r_2),(d_1 \textit{ subsumes } d_2)}{((\mathsf{rel}\ r_1\ (\mathsf{num}\ ((\mathsf{at\text{-}least}\ x_1),d_1)))\ \textit{subsumes}\ (\mathsf{rel}\ r_2\ (\mathsf{num}\ ((\mathsf{at\text{-}least}\ x_2),d_2)))}$$
  where $x_1 \leq x_2$.

  Soundness Proof: If $d \in \mu((\mathsf{rel}\ r_2\ (\mathsf{num}\ ((\mathsf{at\text{-}least}\ x_2),d_2))))$ then $\left|\{\hat{d} \in \mathcal{D} \mid (\hat{d} \in \mu(d_2)) \wedge (d,\hat{d}) \in \mu(r_2)\}\right| \geq x_2$. Since ($d_1$ *subsumes* d_2(and (r_1 *subsumes* r_2), $\{\hat{d} \in \mathcal{D} \mid (\hat{d} \in \mu(d_2)) \wedge (d,\hat{d}) \in \mu(r_2)\} \subseteq \{\hat{d} \in \mathcal{D} \mid (\hat{d} \in \mu(d_1)) \wedge (d,\hat{d}) \in \mu(r_1)\}$ and thus $\left|\{\hat{d} \in \mathcal{D} \mid (\hat{d} \in \mu(d_1)) \wedge (d,\hat{d}) \in \mu(r_1)\}\right| \geq x_2 \geq x_1$. Thus $d \in \mu((\mathsf{rel}\ r_1\ (\mathsf{num}\ ((\mathsf{at\text{-}least}\ x_1),d_1))))$, so $\mu((\mathsf{rel}\ r_1\ (\mathsf{num}\ ((\mathsf{at\text{-}least}\ x_1),d_1)))) \subseteq \mu((\mathsf{rel}\ r_2\ (\mathsf{num}\ ((\mathsf{at\text{-}least}\ x_2),d_2))))$.

- $$\frac{(r_2 \textit{ subsumes } r_1),\ (d_2 \textit{ subsumes } d_1)}{((\mathsf{rel}\ r_1\ (\mathsf{num}\ ((\mathsf{at\text{-}most}\ x_1),d_1)))\ \textit{subsumes}\ (\mathsf{rel}\ r_2\ (\mathsf{num}\ ((\mathsf{at\text{-}most}\ x_2),d_2))))}$$
  where $x_1 \geq x_2$.

  Soundness Proof: If $d \in \mu((\mathsf{rel}\ r_2\ (\mathsf{num}\ ((\mathsf{at\text{-}most}\ x_2),d_2))))$ then $\left|\{\hat{d} \in \mathcal{D} \mid (\hat{d} \in \mu(d_2)) \wedge (d,\hat{d}) \in \mu(r_2)\}\right| \leq x_2$. Since ($d_2$ *subsumes* d_1) and (r_2 *subsumes* r_1), $\{\hat{d} \in \mathcal{D} \mid (\hat{d} \in \mu(d_1)) \wedge (d,\hat{d}) \in \mu(r_1)\} \subseteq \{\hat{d} \in \mathcal{D} \mid (\hat{d} \in \mu(d_2)) \wedge (d,\hat{d}) \in \mu(r_2)\}$ and thus $\left|\{\hat{d} \in \mathcal{D} \mid (\hat{d} \in \mu(d_1)) \wedge (d,\hat{d}) \in \mu(r_1)\}\right| \leq x_2 \leq x_1$. Thus $d \in \mu((\mathsf{rel}\ r_1\ ((\mathsf{num}\ (\mathsf{at\text{-}least}\ x_1),d_1))))$, so $\mu((\mathsf{rel}\ r_1\ (\mathsf{num}\ ((\mathsf{at\text{-}least}\ x_1),d_1)))) \subseteq \mu((\mathsf{rel}\ r_2\ (\mathsf{num}\ ((\mathsf{at\text{-}least}\ x_2),d_2))))$.

- $$\frac{}{((\mathsf{rel}\ r\ (\mathsf{num}\ ((\mathsf{from}\ x_{1_b}\ \mathsf{to}\ x_{1_t}),d)))\ \textit{subsumes}\ (\mathsf{rel}\ r\ (\mathsf{num}\ ((\mathsf{from}\ x_{2_b}\ \mathsf{to}\ x_{2_t}),d))))}$$
  if $x_{1_t} \geq x_{2_t}$ and $x_{1_b} \leq x_{2_b}$

  Soundness Proof: If $d \in \mu((\mathsf{rel}\ r\ (\mathsf{num}\ ((\mathsf{from}\ x_{2_t}\ \mathsf{to}\ x_{2_b}),d))))$ then $x_{2_b} \leq \left|\{\hat{d} \in \mathcal{D} \mid (\hat{d} \in \mu(d_2)) \wedge (d,\hat{d}) \in \mu(r_2)\}\right| \leq x_{2_t}$. So clearly $x_{1_b} \leq \left|\{\hat{d} \in \mathcal{D} \mid (\hat{d} \in \mu(d_2)) \wedge (d,\hat{d}) \in \mu(r_2)\}\right| \leq x_{1_t}$. Thus $d \in \mu((\mathsf{rel}\ r\ (\mathsf{num}\ ((\mathsf{from}\ x_{1_t}\ \mathsf{to}\ x_{1_b}),d)))$ and therefore $\mu((\mathsf{rel}\ r\ (\mathsf{num}\ ((\mathsf{from}\ x_{1_t}\ \mathsf{to}\ x_{1_b}),d)))) \subseteq \mu((\mathsf{rel}\ r\ (\mathsf{num}\ ((\mathsf{from}\ x_{2_t}\ \mathsf{to}\ x_{2_b}),d))))$.

Although there are a number of different ways to construct an interval in addition to the ones specified above, they can be trivially reduced to one of the cases already presented. In the language we allow the user to specify any number of `numeric-rel-op`'s, and we can convert the user input to a canonical form which will fall into one of three categories: greater than x, less than x, or between two values. The other numeric operators which we have not included above are **less-than** and **more-than** which are the same as **at-least** and **at-most** but with $<$ instead of $\leq$ and $>$ in place of $\geq$.

In addition to the above, there is a simply interaction between statements and modifier. The following relationships closely parallel the subsumption of modifiers described above and we omit most of the proofs since they are almost identical to the proceeding set:

- $\frac{(\textsf{(every } d_3) \ r_2 \ \textsf{(only } d_2)), \ (r_2 \ \textit{subsumes} \ r_1),(d_1 \ \textit{subsumes} \ d_2)}{(\textsf{(rel } r_1 \ \textsf{(only } d_1)) \ \textit{subsumes} \ d_3)}$

  Soundness Proof: If $d \in \mu(d_3)$ then by the table in section 3.2.6, $\forall \hat{d} \in \mathcal{D}$ such that $(d, \hat{d}) \in \mu(r_2)$ we know that $\hat{d} \in \mu(d_2)$. Since $\mu(r_1) \subseteq \mu(r_2)$ we know that $\{\hat{d} \in \mathcal{D} \mid (d, \hat{d}) \in \mu(r_1)\} \subseteq \{\hat{d} \in \mathcal{D} \mid (d, \hat{d}) \in \mu(r_2)\}$. Further since $(d_1$ *subsumes* $d_2)$ it is clear that if $\hat{d} \in \mu(d_2)$ then $\hat{d} \in \mu(d_1)$. If $\forall \hat{d} \in \mathcal{D}$ s.t. $(d, \hat{d}) \in \mu(r_2)$ implies $\hat{d} \in \mu(d_2)$ then $\forall \hat{d} \in \mathcal{D}$ s.t. $(d, \hat{d}) \in \mu(r_1)$ implies $\hat{d} \in \mu(d_1)$ since both the condition and the implication are weaker in the second statement. Thus $d \in \mu(\textsf{(rel } r_1 \ \textsf{(only } d_1)))$ and hence $\mu(d_3) \subseteq \mu(\textsf{(rel } r_2 \ \textsf{(only } d_2)))$.

- $\frac{(\textsf{(every } d_3 \ r_2 \ \textsf{(every } d_2)), \ (d_2 \ \textit{subsumes} \ d_1),(r_1 \ \textit{subsumes} \ r_2)}{(\textsf{(rel } r_1 \ \textsf{(every } d_1)) \ \textit{subsumes} \ d_3)}$

- $\frac{(\textsf{(every } d_3) \ r_2 \ \textsf{(some } d_2)), \ (d_1 \ \textit{subsumes} \ d_2),(r_1 \ \textit{subsumes} \ r_2)}{(\textsf{(rel } r_1 \ \textsf{(some } d_1)) \ \textit{subsumes} \ d_3)}$

- $\frac{(\textsf{(every } d_3) \ r_2 \ \textsf{(no } d_2)),(d_2 \ \textit{subsumes} \ d_1),(r_2 \ \textit{subsumes} \ r_1)}{(\textsf{(rel } r_1 \ \textsf{(no } d_1)) \ \textit{subsumes} \ d_3)}$

- $\frac{(\textsf{(every } d_3) \ r_2 \ \textsf{(qty } d_2)), \ (d_1 \ \textit{subsumes} \ d_2),(r_1 \ \textit{subsumes} \ r_2)}{(\textsf{(rel } r_1 \ \textsf{(qty } d_1)) \ \textit{subsumes} \ d_3)}$

- $\frac{\{(\textsf{(every } d_3) \ r_2 \ \textsf{(num ((at-least } x_2),d_2))),(r_1 \ \textit{subsumes} \ r_2),(d_1 \ \textit{subsumes} \ d_2)}{(\textsf{(rel } r_1 \ \textsf{(num ((at-least } x_1),d_1))) \ \textit{subsumes} \ d_3)}$

  where $x_1 \leq x_2$.

- $\frac{\{(\textsf{(every } d_3) \ r_2 \ \textsf{(num ((at-most } x_2),d_2))), \ (r_2 \ \textit{subsumes} \ r_1),(d_2 \ \textit{subsumes} \ d_1)}{(\textsf{(rel } r_1 \ \textsf{(num ((at-most } x_1),d_1))) \ \textit{subsumes} \ d_3)}$

  where $x_1 \geq x_2$.

- $\dfrac{\{((\textsf{every}\ d_3)\ r\ (\textsf{num}\ ((\textsf{from}\ x_{2_{bottom}}\ \textsf{to}\ x_{2_{top}}),d)))}{((\textsf{rel}\ r\ (\textsf{num}\ ((\textsf{from}\ x_{1_{bottom}}\ \textsf{to}\ x_{1_{top}}),d)))\ \textit{subsumes}\ d_3)}$
  if $x_{1_{top}} \geq x_{2_{top}}$ and $x_{1_{bottom}} \leq x_{2_{bottom}}$.

The final element of this discussion involves subsumption across different quantificational tags. We will state the following theorems without proof although they are all straightforward:

- $\dfrac{}{(\textsf{rel}\ r\ (\textsf{one}\ d))\ \textit{subsumes}\ (\textsf{rel}\ r\ (\textsf{num}\ ((\textsf{exactly}\ 1),d)))}$
- $\dfrac{}{(\textsf{rel}\ r\ (\textsf{no}\ d))\ \textit{subsumes}\ (\textsf{rel}\ r\ (\textsf{num}\ ((\textsf{at-most}\ 0),d)))}$
- $\dfrac{}{(\textsf{rel}\ r\ (\textsf{some}\ d))\ \textit{subsumes}\ (\textsf{rel}\ r\ (\textsf{every}\ d))}$
- $\dfrac{}{(\textsf{rel}\ r\ (\textsf{some}\ d))\ \textit{subsumes}\ (\textsf{rel}\ r\ (\textsf{one}\ d))}$

3.4.4 Relations

- $\dfrac{(r_1\ \textit{subsumes}\ r_2)}{((\textsf{inverse}\ r_1)\ \textit{subsumes}\ (\textsf{inverse}\ r_2))}$
  Soundness Proof: If $(d_a, d_b) \in \mu((\textsf{inverse}\ r_2))$ therefore $(d_b, d_a) \in \mu(r_2)$. Hence if $(r_1\ \textit{subsumes}\ r_2)$ then $(d_b, d_a) \in \mu(r_1)$ and $(d_a, d_b) \in \mu(\ (\textsf{inverse}\ r_1))$. Thus $\mu((\textsf{inverse}\ r_1)) \subseteq \mu((\textsf{inverse}\ r_2))$.

- $\dfrac{(r_2\ \textit{subsumes}\ r_1)}{((\textsf{all}\ r_1)\ \textit{subsumes}\ (\textsf{all}\ r_2))}$
  Soundness Proof: If $(d_a, d_b) \in \mu((\textsf{all}\ r_2))$ then $\{d \in \mathcal{D} \mid (d_a, d) \in \mu(r_2)\} \subseteq \mu(d_b)$. Thus if $(r_2\ \textit{subsumes}\ r_1)$ then $\{d \in \mathcal{D} \mid (d_a, d) \in \mu(r_1)\} \subseteq \{d \in \mathcal{D} \mid (d_a, d) \in \mu(r_2)\} \subseteq \mu(d_b)$. Hence, $(d_a, d_b) \in \mu((\textsf{all}\ r_1))$ so $\mu((\textsf{all}\ r_2)) \subseteq \mu((\textsf{all}\ r_1))$.

- $\dfrac{\{(r_1\ \textit{subsumes}\ \hat{r}_1), (r_2\ \textit{subsumes}\ \hat{r}_2), \ldots (r_n\ \textit{subsumes}\ \hat{r}_n)}{((\textsf{chain}\ r_1\ r_2 \ldots r_n)\ \textit{subsumes}\ (\textsf{chain}\ \hat{r}_1\ \hat{r}_2 \ldots \hat{r}_n))}$
  Soundness Proof: If $(d_a, d_b) \in \mu((\textsf{chain}\ \hat{r}_1\ \hat{r}_2 \ldots \hat{r}_n))$ then $\exists d_1, d_2, \ldots, d_n$ such that $(d_a, d_1) \in \mu(\hat{r}_1), (d_1, d_2) \in \mu(\hat{r}_2), \ldots, (d_n, d_b) \in \mu(\hat{r}_n)$. So if $(r_i\ \textit{subsumes}\ \hat{r}_i)$ for $1 \leq i \leq n$ then $(d_a, d_1) \in \mu(r_1), (d_1, d_2) \in \mu(r_2), \ldots, (d_n, d_b) \in \mu(r_n)$. Thus $(d_a, d_b) \in \mu((\textsf{chain}\ r_1\ r_2 \ldots r_n)$, so $\mu((\textsf{chain}\ r_1\ r_2 \ldots r_n) \subseteq \mu((\textsf{chain}\ \hat{r}_1\ \hat{r}_2 \ldots \hat{r}_n))$.

- $\dfrac{(r_1\ \textit{subsumes}\ r_2),(d_1\ \textit{subsumes}\ d_2)}{(\textsf{constrain}\ r_1\ d_1)\ \textit{subsumes}\ (\textsf{constrain}\ r_2\ d_2)}$
  Soundness Proof: If $(d_a, d_b) \in \mu((\textsf{constrain}\ r_2\ d_2))$ then $(d_a, d_b) \in \mu(r_2)$ and

$d_b \in \mu(d_2)$. So if (r_1 *subsumes* r_2) and (d_1 *subsumes* d_2) then $(d_a, d_b) \in \mu(r_1)$ and $d_b \in \mu(d_1)$. Hence $(d_a, d_b) \in \mu((\textsf{constrain}\ r_1\ d_1))$ so $\mu((\textsf{constrain}\ r_1\ d_1)) \subseteq \mu((\textsf{constrain}\ r_2\ d_2))$.

3.5 Soundness and Completeness

We will now prove the soundness of our system with respect to our semantics, and conjecture that completeness also follows. However, before we can do that we need to state the form of a proof using the inference relation. A knowledge base consists of a collection of statements and we define KB $\vdash s$, read as s is deducible from the knowledge base, to be true if and only if there exists a series of statements $s_0, s_1, \ldots, s_n$ where $s_n = s$, such that for each s_i, either $s_i \in$ KB, or $\vdash s_i$, or there exists some inference rule $\mathbf{R}$ listed in section 3.4.1 to section 3.4.4 and some set $\{\hat{s}_0, \hat{s}_1, \ldots, \hat{s}_m\}$ such that for all j, where $0 \leq j \leq m < i$, $\hat{s}_j = s_k$ for some k, $0 \leq k \leq i$, and $\{\hat{s}_0, \hat{s}_1, \ldots, \hat{s}_m\} \vdash_{\mathbf{R}} s_i$.

Theorem 1 (Soundness) *For any two descriptions d_1, d_2 and a given knowledge base, KB, if KB $\vdash$ (d_1 satisfies d_2) then KB $\models$ (d_1 satisfies d_2).*

Proof: If KB $\vdash$ (d_1 *satisfies* d_2) then it follows directly from our notion of a proof using $\vdash$ that the statement (d_1 *satisfies* d_2) must follow as the result of the application of a finite number of rules using the statements in the KB. Since we have already proved all of these rules sound with respect to the semantics, it follows that the repeated application of these rules would also be sound.

Conjecture 1 (Completeness) *For any two descriptions d_1, d_2 and a given knowledge base, KB, if KB $\models$ (d_1 satisfies d_2) then KB $\vdash$ (d_1 satisfies d_2).*

Unfortunately, we expect a proof of completeness would be considerably more complex than the soundness proof, and although we believe that it is provable, we will consider the proof to be the subject of future work.

Chapter 4

Future Work and Conclusions

Our view of Woods' technical report is that to a large degree it is an important and careful advocacy for a different methodology in the designing of general purpose knowledge representation systems. For example he argues at some length that previous tractability research has been using an inappropriate metric in judging KL-ONE style systems. However, although Woods puts forward a framework for tackling a number of issues that have bedeviled the KR community for years, the details of the framework often are very sketchy, and Woods' contribution to some degree can be summarized as an exhortation that the issues can in fact be successfully resolved if the community would take a less restrictive approach. This work involves an effort to further specify several of these important details.

To a large degree Woods' has two central claims, namely that effective KR systems should provide:

- Quantificational tags allowing links to carry explicit quantificational information; and
- A notion of subsumption which is based on the intensional meaning of conceptual structures rather than the extensional meaning which has dominated most previous KL-ONE style systems.

This thesis took these two claims as its focus, and we summarize our conclusions about each in turn.

4.1 Quantificational Tags

Woods' argues that quantificational tags allow the knowledge engineer to make explicit much of the information that has been left implicit in other KL-ONE style systems and keeping this information implicit causes significant confusion both for the engineer and for the end user. The classic example from Woods' 1975 paper [29] of a link connecting the concepts black and telephone provides a key illustration of the need for explicit quantificational information. Among the possible meanings of the link between the concepts black and telephone are: that all telephones must be black, that there exists at least one telephone which is black, that there exists only one black telephone, that every black thing is a telephone, or that some black thing is a telephone.

Although Woods' 1975 criticisms in [29] were taken to heart by people in the semantic nets community, who then worked harder to better define the semantics of their links, a single choice of meanings from the above list in all cases still is not adequate and in practice the user needs to be able to specify explicitly which one of the above meanings fits each particular case. The syntax and semantics put forth by Woods in [31] provides a moderately concrete mechanism for allowing this.

Our contribution in this area has been to incorporate Woods' tagging system into a larger framework. We have generalized Woods' tags somewhat by incorporating several additional tags and combinations of tags beyond those mentioned by Woods, and we have developed a syntax we believe will be more straightforward and intuitive to the naive reader. In addition we have used the framework of quantificational tags to provide a natural way for having a numerical value or interval be the object of a link, an important extension in light of previous work representing medical knowledge which demonstrated the critical role of the ability to reason with numerical values as the object of a given relation.

4.2 Intensional Foundations

The focus which Woods brings to an intensional rather than extensional semantics is in many ways the most important foundational issue that he raises. In extensional subsumption one concept is considered to subsume another if and only if every thing which satisfies the latter concept also satisfies the former concept. Although extensional subsumption has generally been accepted without much debate for most KL-ONE-like systems, this form of subsumption has two major drawbacks. The first has to do with the computational complexity. For example, in an extensional system one is required to determine that the concept of a language determined by a context free grammar and the concept of a language accepted by a pushdown automata subsume each other, presuming that these concepts could even be represented. The burden of recognizing such coincidences is very significant for the inferential mechanism. In fact, it turns out that subsumption even in fairly restricted languages is often intractable if not undecidable. The other major problem with extensional subsumption is that it violates some fundamental intuitions regarding identifying concepts with empty extensions (e.g. unicorns, 150 year old men, etc.).

In response Woods argues that this notion of subsumption is in reality a conflation of five distinct senses of subsumption and that rather than making extension subsumption the basis for an inference system that in its place we should substitute structural subsumption in which, "the subsuming concept is determined to be more general than the subsumed concept by virtue of the formally specified subsumption criterion applied to the structure of the descriptions – preferably by an algorithm that is computational tractable or at least more efficient than general deduction." In addition Woods suggests that it should be possible to make statements about concepts themselves rather than being required to only refer to their extensions, and hence this allows one to in some ways view a concept both as an "individual" and a "class".

Unfortunately, Woods did not discuss how to set up the semantics underneath this new notion of a conceptual description nor did he describe what exactly would be the formal notion of subsumption, and it is the task of filling in those foundations that is the core contribution of this work. We describe how to create and structure the domain of conceptual descriptions over which the system reasons and how to assign meaning the various language constructs in order to allow statements to be made about concepts themselves rather than their referents. Furthermore, we specify the details of an inference relation to determine subsumption relationships which is sound with respect to our semantics and we believe also complete, although that is the subject of future work.

Another item for future work is that Woods' suggests that one might want the option to have a concept be satisfied *in* a given situation as a notion of satisfaction with regard to conceptual descriptions that goes beyond the standard predicate calculus notion of a concept being satisfied *by* some entity. Our notion of contexts opens the possibility of providing for concepts which are satisfied only within certain contexts, but in this work we really only deal with the notion of a concept being satisfied by a given entity.

4.3 Conclusion

In summary this thesis provides further development along one of the more promising lines of general purpose knowledge representation research. In the early days of semantics networks, it was possible to represent a large variety of concepts but because of the absence of explicit semantics even the authors of such systems were sometimes contradictory or easily confused in their usage, and different users would inevitably interpret the structures differently from how the designers had intended. As a result the field moved sharply in the direction of systems with very precise, rigid semantics based on first order logic and adopted a set of goals (soundness and completeness) which so drastically curtailed expressivity that most real world knowledge fell notable out of reach of these limited system.

In his work, Woods advocates abandoning some of the more rigid requirements of the KRYPTON generation (e.g. completeness) and distances himself from the notion of a concept as a first order predicate. He also advances the idea of intensional subsumption as a way to deal with a number of difficult computation issues while providing a more intuitive basis for subsumption, and presents the idea of quantificational tags as a means to enhance expressivity and deal with a significant source of past confusion. In this thesis, we take Woods' sketch as a starting point and construct a KR language with a fully specified set of syntax and semantics. Most importantly, we take Woods' notions of "conceptual descriptions" and "satisfaction" and provide a firm set of formal foundations which are congruent to the demands of the remained Woods' framework.

Clearly this work represents an intermediate step in the path to more powerful and useful knowledge representation systems, however given the history of KR it seems that considerable investments of time and effort into the design of the foundations of the central representational constructs is essential to being able to make important distinction later down the line. We are optimistic that future real-world applications we be able to rest solidly on the foundations advance here.

Bibliography

[1] C. Alchourrón, P. Gärdenfors, and D. Makinson. On the logic of theory change: Partial meet contraction functions and their associated revision functions. *Journal of Symbolic Logic*, 50:510–530, 1985.

[2] Kenneth L. Becker, editor. *Principles and Practice of Endocrinology and Metabolism.* J.B. Lippincott Company, Grand Rapids, 1990.

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

[4] Ronald J. Brachman, Richard E. Fikes, and Hector J. Levesque. KRYPTON: a functional approach to knowledge representation. In Ronald J. Brachman and Hector J. Levesque, editors, *Readings in Knowledge Representation*, chapter 6, pages 411–429. Morgan Kaufmann Publishers, Inc., Los Altos, California, 1985.

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

[6] Jon Doyle and Ramesh S. Patil. Two theses of knowledge representation: Language restrictions, taxonomic classification, and the utility of representation services. *Artificial Intelligence*, 48(3):261–297, April 1991.

[7] Ira J. Haimowitz. Using NIKL in a large medical knowledge base. TM 348, Massachusetts Institute of Technology, Laboratory for Computer Science, 545 Technology Square, Cambridge, MA, 02139, January 1988.

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

[9] Graeme J. Hirst. Ontological assumptions in knowledge representation. In Ronald J. Brachman, Hector J. Levesque, and Raymond Reiter, editors, *First Internation Conference on Principle of Knowledge Representation*, pages 157–169, San Mateo, 1989. Morgan Kaufmann, Inc.

[10] Thomas S. Kaczmarek, Raymond Bates, and Gabriel Robins. Recent developments in NIKL. In *Proceedings of the National Conference on Artificial Intelligence*, pages 978–985. American Association for Artificial Intelligence, 1986.

[11] Hector J. Levesque and Ronald J. Brachman. A fundamental tradeoff in knowledge representation and reasoning (revised version). In Ronald J. Brachman and Hector J. Levesque, editors, *Readings in Knowledge Representation*, pages 41–70. Morgan Kaufmann, 1985.

[12] Hector J. Levesque and Ronald J. Brachman. Expressiveness and tractability in knowledge representation and reasoning. *Computational Intelligence*, 3:78–93, 1987.

[13] R.M. MacGregor. The evolving technology of classification-based knowledge representations systems. In John Sowa, editor, *Principles of semantics networks: explorations in the representation of knowledge.* Morgan Kaufmann, 1991.

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

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

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

[17] Bernhard Nebel. Terminological reasoning is inherently intractable. *Artificial Intelligence*, 43(2):235–249, May 1989.

[18] Bernhard Nebel. Terminological cycles: Semantics and computational properties. In John Sowa, editor, *Principles of semantics networks: explorations in the representation of knowledge*. Morgan Kaufmann, 1991.

[19] Peter Patel-Schneider. Undecidability of subsumption in NIKL. *Artificial Intelligence*, 39(2):263–272, April 1989.

[20] R. S. Patil, P. Szolovits, and W. B. Schwartz. Information acquisition in diagnosis. In *Proceedings of the National Conference on Artificial Intelligence*, pages 345–348. American Association for Artificial Intelligence, 1982.

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

[22] Burton D. Rose. *Clinical Physiology of Acid-Base and Electrolyte Disorders.* McGraw-Hill Information Services Company, New York, 1989.

[23] Manfred Schmidt-Schauss. Subsumption in KL-ONE undecidable. In Ronald J. Brachman, Hector J. Levesque, and Raymond Reiter, editors, *Proceedings of the First International Conference on Principles of Knowledge Representation and Reasoning (KR'89)*, pages 421–431, Toronto, Ontario, 1989.

[24] James G. Schmolze and Thomas 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.

[25] James G. Schmolze and William S. Mark. The nikl expeperience. *Computational Intelligence*, 7(1):48=69, February 1991.

[26] Brian 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.

[27] Stephen W. Smoliar and William Swartout. A report from the frontiers of knowledge representation. Technical Report ISI/RR-90-283, USC Information Sciences Institute, 4676 Admiralty Way, Marina del Rey, California 90291, 1990.

[28] Marc 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.

[29] William A. Woods. What's in a link: Foundations for semantic networks. In Bobrow and Collins, editors, *Representation and Understanding*, pages 35–82. Academic Press, New York, 1975.

[30] William A. Woods. Important issues in knowledge representation. In *Proceedings of the IEEE Vol-74*, 1986.

[31] William A. Woods. Understanding subsumption and taxonomy: A framework for progress. Technical Report TR-19-90, Center for Research in Computing Technology, Harvard University, Cambridge, MA 02138, 1990. Also appears in *Principles of Semantic Networks*, edited by John F. Sowa.