Terminological Representation, Natural Language & Relation Algebra

Renate A. Schmidt

Max-Planck-Institut für Informatik, Im Stadtwald
W-6600 Saarbrücken 11, Germany

Abstract. In this paper I establish a link between KL-ONE-based knowledge representation concerned with *terminological representation* and the work of P. Suppes (1976, 1979, 1981) and M. Böttner (1985, 1989) in computational linguistics. I show how this link can be utilised for the problem of finding adequate terminological representations for given information formulated in ordinary English.

1 Introduction

KL-ONE (Brachman and Schmolze 1985) is a knowledge representation system that R. J. Brachman started developing in the mid-seventies. KL-ONE has many variants. Examples are KRYPTON (Brachman, Gilbert and Levesque 1985), NIKL (Schmolze and Mark 1991), LOOM (MacGregor 1991), BACK (Nebel and von Luck 1988), CLASSIC (Borgida, Brachman, McGuinness and Resnick 1989) and $\mathcal{KRIS}$ (Baader and Hollunder 1991). Early research focussed on providing formal syntactic and semantic definitions for the different systems (in accordance with the criticism of Woods (1975), Hayes (1977) and McDermott (1978)). This led to the discovery that inference in NIKL (Patel-Schneider 1989) and KL-ONE (Schmidt-Schauß 1989) is undecidable. The debate on the tradeoff between tractability of inference and expressiveness of language in, e.g., Levesque and Brachman (1987) and Doyle and Patil (1991) initiated the analysis of the computational complexity of the family of *attributive concept description languages*, also called $\mathcal{AL}$-languages. The investigations of Schmidt-Schauß and Smolka (1991), Donini, Lenzerini, Nardi and Nutt (1991) and also Nebel (1988), among others, give some indication on the effect that including different syntactic operators in a representation language has on the computational cost of inference. These investigations show that tractability can only be achieved at the cost of considerably reducing the expressiveness of the representation language. However, expressively limited systems don't seem to be very useful in practice. According to Doyle and Patil (1991) and Schmolze and Mark (1991) users tend to find expressive but intractable systems more useful than systems that are tractable but expressively impoverished. Especially Doyle and Patil (1991) argue against the bias towards computational tractability. Instead they argue in favour of expressiveness of language. After all, usually a domain of application is specified in ordinary natural language. And natural language is expressively very powerful.

Reflecting on the history of knowledge representation and discussion future directions in research Brachman (1990) notes that natural language specific issues have received less attention than they should have. He says (p. 1091):

> It would not hurt at this point to go back and spend some time thinking about the relation of KR [knowledge representation] to natural language, for example—after all, that was in part responsible for the birth of the field in the first place.

In this paper I do just that. I relate terminological representation to natural language and show how the work of P. Suppes (1976, 1979, 1981) and M. Böttner (1985, 1989) in computational linguistics can be utilised.

The setting of my discussion is an algebraic one. In Brink and Schmidt (1992), Schmidt (1991) and Brink, Britz and Schmidt (1992) we showed that terminological representation languages can be interpreted algebraically. The algebras we use are Tarski's (1941) *relation algebras*, Brink's (1981) *Boolean modules* and new algebras called *Peirce algebras*. These algebras are closely related to the algebras Suppes (1976) uses in his semantic analysis of a fragment of the English language. Suppes presents a system for translating natural language phrases and sentences as relation algebraic expressions. As these can be associated with terminological expressions I propose that the work of Suppes and also Böttner be taken as a formal basis for finding adequate terminological representations for domain information formulated in English, and vice versa, for finding the English formulations for terminological expressions.

In Section 2 I give a brief account of the relation algebraic semantics of terminological representation languages as presented in Brink and Schmidt (1992), Brink et al. (1992) and Schmidt (1991). In Section 3 I outline the relation algebraic analysis of the English language by Suppes (1976, 1979, 1981) and Böttner (1985, 1989). And in Section 4 I show by way of examples how their analysis is relevant to terminological representation.

2 Terminological Representation and Relation Algebra

KL-ONE-based knowledge representation systems have two components: a TBox and an ABox. Each component has its own representation language and inference system. *Terminological representation* is concerned with the TBox which contains information defined as interrelationships among *concepts* or *roles*. Concepts are interpreted as sets and roles as binary relations. In this paper I denote concepts by C and D and roles by R and S. The ABox contains *assertional representations*. It contains information about elements of concepts and roles.

The different terminological representation languages distinguish themselves by the syntactic operators they provide for constructing complex concept and role descriptions. In column one of Figure 1 I list a subset of operators available in the language $\mathcal{U}$ of Patel-Schneider (1987) and the language $\mathcal{KL}$ of Woods and Schmolze (1992). These are very expressive terminological languages. Most terminological languages like the $\mathcal{AL}$-languages provide only a selection of these operators.

Fig. 1. Algebraic Semantics for terminological constructs

Concept-forming operators on concepts		Set-theoretic operations (modelled in Boolean algebra)	
Top concept	$\top$	Universe of interpretation	U
Bottom concept	$\bot$	Empty set	$\emptyset$
Conjunction	(and C D)	Intersection	$C \cap D$
Disjunction	(or C D)	Union	$C \cup D$
Negation	(not C)	Complement	C'
Role-forming operators on roles		**Relation-theoretic operations (modelled in relation algebra)**	
Top role	∇	Universal relation	U^2
Bottom role	$\mathit{\Lambda}$	Empty relation	$\emptyset$
Role conjunction	(and R S)	Intersection	$R \cap S$
Role disjunction	(or R S)	Union	$R \cup S$
Role negation	(not R)	Complement	R'
Identity role	self	Identity relation	Id
Inversion	(inverse R)	Converse	$R^{\smile}$
Composition	(compose R S)	Relational composition	$R\,;S$
Concept-forming operators on roles		**Set-forming operations on relations (modelled in Boolean modules)**	
Existential restriction	(some R C)	Peirce product	$R:C$
Universal restriction	(all R C)	Involution	$(R:C')'$
Role-forming operators on concepts		**Relation-forming operations on sets (modelled in Peirce algebra)**	
Role restriction	(restrict R C)	Range restriction	$R \rfloor C$

Concepts can be interrelated by the *subsumption* relation. Subsumption is interpreted as the subset relation (or, depending on the point of view as the superset relation). I write C $\sqsubseteq$ D if C is subsumed by D. Concepts can also be defined to be *equivalent* or *disjoint*. Two concepts C and D are said to equivalent, written C $\doteq$ D, if they mutually subsume each other. Disjoint concepts C and D, specified with (disjoint C D), are interpreted as disjoint sets. Interrelating concepts by subsumption, equivalence and disjointness is often limited in some way. The left hand side of a subsumption or equivalence specification is commonly restricted to be a *primitive* concept, that is, a concept not defined as a compound concept term. Also, usually only *primitive* concepts can be defined to be disjoint. For roles subsumption, equivalence and disjointness are defined similarly.

I now discuss the algebraic semantics. An *interpretation* of a terminological representation language is given as usual by a pair $(U, \cdot^{\mathcal{I}})$. U is the *domain of interpretation* and $\cdot^{\mathcal{I}}$ is the *interpretation function*, mapping any concept C to a subset of U (i.e., to an element in $\mathbf{2}^U$, the powerset of U) and any role to a binary relation over U (i.e., to an element in $\mathbf{2}^{U^2}$). I abbreviate $\mathsf{C}^{\mathcal{I}}$ and $\mathsf{R}^{\mathcal{I}}$ by C and R, respectively. Now, instead of defining the meaning of the terminological operators model-theoretically (in terms of first-order logic), as it is usually done, Brink and Schmidt (1992) and Brink et al. (1992) show that the semantics can be equivalently defined in terms of algebraic operations. The alternative algebraic semantics is given in the second column of Figure 1 in which each terminological operator is associated with a set-theoretic operation. The Figure is subdivided according to the different kinds of operators with which new concepts and roles arise from primitive ones:

Concept-forming operators on concepts: These include the designated concepts *top* and *bottom* (here regarded as nullary operators) and the *conjunction*, *disjunction*, and *negation* operators. Each is associated with a set-theoretic constant or operation. Namely, the top concept with the domain of interpretation U, the bottom concept with the empty set $\emptyset$, conjunction with intersection $\cap$, disjunction with union $\cup$ and negation with complementation $'$ taken with respect to U. Just as the set-theoretic operations are characterised in *Boolean algebras*, their corresponding terminological operators are also characterised in Boolean algebras. For the set of concepts is partially ordered with respect to the *subsumption* relation in a *concept taxonomy*. If each pair of concepts has a meet and a join (that is, both their conjunction and disjunction exist) the concept taxonomy forms a lattice. It forms a Boolean algebra if each concept has a complement (that is, the negation exists) and meet and join distribute over each other.

The conjunction operator is available in most terminological systems. But few terminological representation systems provide also the disjunction and negation operators.

Role-forming operators on roles: Even fewer terminological representation languages provide conjunction, disjunction and negation also for roles. As for concepts the Boolean operators on roles are interpreted by their Boolean counterparts, this time applied to binary relations. Other operators on roles forming new roles are *inverse* and *composition*. For these, the respective relation-theoretic counterparts are the *converse* relation (denoted $\smile$) and *relational composition* (denoted $;$). Given two binary relations R and S the converse of R is defined by

(1) $\quad R^{\smile} = \{(x, y) \mid (y, x) \in R\}$

and the composition of R and S is given by

(2) $\quad R \,;S = \{(x, y) \mid (\exists z)[(x, z) \in R \;\&\; (z, y) \in S]\}\,.$

The designated role self has the *identity relation* Id over U as relation-theoretic counterpart. The characterising algebra for binary relations interacting in this way is Tarski's (1941) *relation algebra*. A relation algebra is a Boolean algebra endowed with a nullary operation (the identity), a unary operation (the converse operation) and a binary operation (the composition) satisfying certain equational axioms. A formal definition of relation algebra can be found in introductory material by Jónsson (1982) and Maddux (1991a, 1991b). Relation algebras provide then also a characterisation of role-forming operators on roles.

Concept-forming operators on roles: New concepts also arise through interactions with roles. The commonly available operators of this kind are *existential restriction* some and *universal restriction* all. (Commonly used alternative notations for (some R C) and (all R C) are $\exists\mathsf{R}:\mathsf{C}$ and $\forall\mathsf{R}:\mathsf{C}$, respectively.) These operators have algebraic versions as well. The algebraic version of the some operator is the *Peirce product* (denoted $:$). Applied to a relation R and a set C the Peirce product yields the set

(3) $R:C = \{x \mid (\exists y)[(x,y) \in R \;\&\; y \in C]\}$.

The algebraic version of the all operator is a variant of Peirce product (called *involution*). Namely:

(4) $(R:C')' = \{x \mid (\forall y)[(x,y) \in R \Rightarrow y \in C]\}$.

Algebras that axiomatise the Peirce product are *Boolean modules*. A Boolean module is a two-sorted algebra of a Boolean algebra and a relation algebra endowed with an operation (corresponding to Peirce product) from the relation algebra to the Boolean algebra satisfying certain equational axioms. A formal definition can be found in Brink (1981) where Boolean modules are introduced. This context also provides an algebraic characterisation for the some and the all operator. Other concept-forming operators on roles that can be accommodated in Boolean modules are *role value maps* and *structural descriptions*. For more details, see Brink et al. (1992) and Schmidt (1991).

Role-forming operators on concepts: Concepts and roles can also be combined to form new roles. An example of such a combination is *role restriction*. Its algebraic version is *range restriction* (denoted $\rfloor$). The restriction of a relation R to a range C is given by:

(5) $R \rfloor C = \{(x,y) \mid (x,y) \in R \;\&\; y \in C\}$.

Algebras formalising this kind of interaction, in particular, range restriction and also role restriction, are *Peirce algebras* which are introduced in Brink et al. (1992). A Peirce algebra is a Boolean module with an additional operation from the Boolean algebra to the relation algebra satisfying certain equational axioms. Other terminological operators that can be accommodated in this context are the operators domain and range of $\mathcal{KL}$ (Woods and Schmolze 1992).

The corresponding relationships to the subsumption ($\sqsubseteq$) and the equivalence ($\doteq$) relations for concepts (respectively for roles) are inclusion and equality in Boolean algebra (respectively in relation algebra). Disjointness corresponds to an inequality of a meet with the zero element (interpreted as the empty set) of the relevant algebra.

3 Natural Language and Relation Algebra

In (1976) and other papers (1973, 1979, 1981) Suppes aims at a systematic analysis of the model-theoretic semantics of fragments of natural language. In Suppes (1979, p. 49) he says:

> The central idea is that the syntax of first-order logic is too far removed from that of any natural language, to use it in a sensitive analysis of the meaning of ordinary utterances.

Instead he proposes an algebraic approach, using so-called *extended relation algebras*. An extended relation algebra $\mathcal{E}(U)$ over a domain U (a non-empty set), is a subset of $\mathbf{2}^U \cup \mathbf{2}^{U^2}$ closed under the operations of union, complementation, converse, relational composition, image and domain restriction. Complementation of sets is taken with respect to U and complementation of relations with respect to U^2. The *image* of a relation R from a set C is the set

(6) $R\,\text{"}\,C = \{y \mid (\exists x)[(x,y) \in R \;\&\; x \in C]\}$,

or equivalently, $R^{\smile} : C$. The *domain restriction* of a relation R to a set C is given by $R^{\smile} \rfloor C$. Extended relation algebras are of model-theoretic nature. As both image and domain restriction can be expressed with Peirce product and range restriction extended relation algebras provide standard models for Peirce algebras.

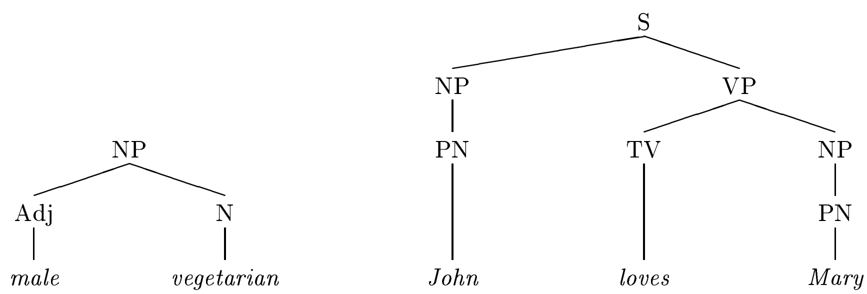
With extended relation algebras Suppes characterises the semantics of English language sentences and phrases. The syntax of natural language is defined by a *phrase structure grammar* G. A grammar is specified in terms of a set of production rules like, for example:

(7)
(i) S $\longrightarrow$ NP + VP
(ii) VP $\longrightarrow$ TV + NP
(iii) NP $\longrightarrow$ Adj + N
(iv) NP $\longrightarrow$ PN.

The symbols S, NP, VP, TV, Adj, N and PN denote 'start symbol', 'noun phrase', 'verb phrase', 'transitive verb', 'adjective', 'noun' and 'proper noun', respectively. Accordingly the syntactic structure of the phrase *male vegetarian* and the sentence *John loves Mary* are represented by the respective syntactic *derivation trees* of Figures 2.

Suppes defines the semantics in two steps. First, he extends the grammar G to a so-called (*potentially*) *denoting grammar*. This is done by associating each production rule of G with a semantic function. The denoting grammar then determines the meaning of phrases and sentences. For example, the semantics of the phrase *male vegetarian* and the sentence *John loves Mary* are determined by the following semantic associations of the above production rules.

Fig. 2. Grammatical derivation trees for *male vegetarian* and *John loves Mary*



(8)	Lexical Production Rule	Semantic Association
(i)	S $\longrightarrow$ NP + VP	$[\mathrm{NP}] \subseteq [\mathrm{VP}]$
(ii)	VP $\longrightarrow$ TV + NP	$[\mathrm{TV}] : [\mathrm{NP}]$
(iii)	NP $\longrightarrow$ Adj + N	$[\mathrm{Adj}] \cap [\mathrm{N}]$
(iv)	NP $\longrightarrow$ PN	$[\mathrm{PN}]$

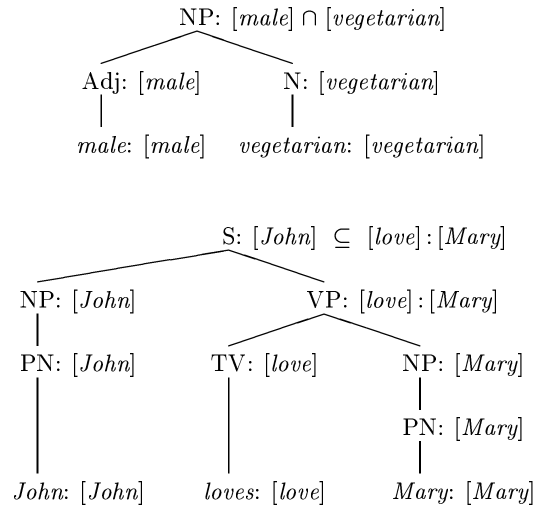
The square brackets indicate the interpretation function. If the adjective *male* is interpreted as the set of male people and the noun *vegetarian* as the set of vegetarians, the intersection $[male] \cap [vegetarian]$ defines the meaning of *male vegetarian*. This can also be read off at the root of the annotated grammatic derivation tree for *male vegetarians* in Figure 3. This tree is called a *semantic tree*. It is derived from the syntactic derivation tree by annotating each node with the appropriate semantic assignment. According to (8) (ii) the verb phrase *loves Mary* is interpreted as the set of lovers of Mary, given by the Peirce product $[love] : [Mary]$. The interpretation $[love]$ of *loves* is a binary relation and the interpretation $[Mary]$ of *Mary* is a singleton set. (In (ii) Suppes actually uses a variant of the image operation, which coincides with Peirce product.) The semantics of the sentence *John loves Mary* is therefore given by:

(9) $[John] \subseteq [love] : [Mary]$.

It is also given by the relevant semantic tree in Figure 3. This illustrates how meaning is assigned to a phrase or sentence by converting its grammatic definition, viewed as a grammatic derivation tree, to a semantic definition, viewed as a semantic tree, via the denotational assignments to the production rules which determine the syntax of the phrase or sentence.

In the second step a *model structure* (U, v) is defined for the phrase structure grammar G. U is any non-empty set regarded as the domain or universe and v, called a *valuation*, is a (partial) function from the vocabulary of terminal symbols in G to the extended relation algebra $\mathcal{E}(U)$. That is, v maps terminal symbols to either sets in $\mathbf{2}^U$ or binary relations in $\mathbf{2}^{U^2}$. As we have seen in the two examples above, nouns and adjectives are mapped to subsets of U, with proper nouns being mapped to singleton sets, and (transitive) verbs are mapped

Fig. 3. Semantic trees for *male vegetarian* and *John loves Mary*



to binary relations.

This algebraic approach has the advantage that it is free of variables and quantifiers over variables. Consequently, according to Suppes (1981, p. 405) the analysis of the semantics of natural language fragments can be carried out directly in English, avoiding the translation into another language (e.g., into the first-order language). Furthermore, it allows the development of a syntactic derivation system for direct inference in the English language. I won't elaborate on this system, but see Suppes (1981).

Since Suppes and Böttner translate English language phrases and sentences as algebraic expressions which, as is evident from Section 2, can be associated with terminological expressions, their work is relevant to the problem of finding adequate terminological representations for information formulated in English and vice versa, for finding English translations for terminological expressions.

In (1976, 1981) Suppes also demonstrates how phrases and sentences with quantifier words (such as *all*, *some* and *no*) in object and subject position are interpreted in the framework of extended relation algebras. For example, the following verb phrases

(10)	Verb phrase	Algebraic interpretation
(i)	*eat all fruit*	$([eat]' : [fruit])'$
(ii)	*eat (some) fruit*	$[eat] : [fruit]$
(iii)	*eat no fruit*	$([eat] : [fruit])'$

are interpreted by variants of the image operation, here appropriately translated as variants of Peirce product. When each of these verb phrases is combined with quantified subjects the semantics of the resulting sentences is of the form similar to that of the sentences

(11)	Sentence	Algebraic interpretation
(i)	*Some persons eat (some) fruit*	$[persons] \cap [eat]:[fruit] \neq \emptyset$
(ii)	*All persons eat (some) fruit*	$[persons] \subseteq [eat]:[fruit]$
(iii)	*No person eats (some) fruit*	$[persons] \cap [eat]:[fruit] = \emptyset$

This gives rise to nine forms of quantified sentences. In (1979) Suppes investigates the algebraic interpretation of negation in verb phrases. The meaning of negated verb phrase like *do not eat (some) fruit* is ambiguous and depends on which word is stressed. For example, if the word *eat* is stressed the interpretation is $[eat]':[fruit]$. Or, if the word *not* is stressed it is $([eat]:[fruit])'$. In (1981) Suppes also defines the semantics of sentences that begin with a demonstrative verb, e.g., *there* as in *There are some birds* and *There are no birds*, and of sentences in which a noun is modified by a relative clause as in *Triangles that cover squares that are projections are isosceles.*

Of particular interest to terminological representation (in particular to the interpretation of the all construct) is the semantics of phrases of the form

(12) *eat only fruit.*

Böttner (1985) interprets this phrase by $[eat]:[fruit] - [eat]:[fruit]'$, or equivalently,

(13) $[eat]:[fruit] \ \cap \ ([eat]:[fruit]')'$.

As Böttner pointed out in (1990), $([eat]:[fruit]')'$ alone inadequately interprets (12). If *eat only fruit* were to be interpreted as $([eat]:[fruit]')'$ one would not be able to deduce that persons who eat only fruit are also persons who eat (some) fruit, since in general

(14) $([eat]:[fruit]')' \not\subseteq [eat]:[fruit]$.

For suppose $[fruit]$ is empty. Then $[eat]:[fruit]$ is empty, but $([eat]:[fruit]')'$ is not necessarily empty, since $([eat]:\emptyset')' = ([eat]:U)' = (\mathrm{dom}([eat]))'$. (For R a relation, $\mathrm{dom}(R)$ denotes the domain of R.) We can show the following:

(15) $([eat]:[fruit]')' \subseteq [eat]:[fruit]$ iff $\mathrm{dom}([eat]) = U$.

But to decree that the domain of each relation must be the entire universe of discourse does not seem feasible. For example, we would not want to include the instances of $[fruit]$ in the domain of $[eat]$. However the interpretation (13) suggested by Böttner is contained in $[eat]:[fruit]$, ensuring that persons eating only fruit also eat some fruit.

In the paper (1985) Böttner not only analyses the semantics of sentences like *John loves only Mary* with *only* in object position, but also of sentences like *Only John loves Mary* and also like *All boys except John love Mary.* In other papers (1989, 1992) he investigates the algebraic interpretation of anaphoric expressions and English imperatives. Examples of anaphoric expressions are *John loves himself*, *John and Mary like each other* and *John likes his toys.* In his most recent work (1991) he also accommodates sentences with verbs in passive form, which he interprets as converse relations.

4 Terminological Representation and Natural Language

Finding adequate terminological representations for the fragment of the English language Suppes and Böttner accommodate in the relation algebraic framework is now straightforward. Words and phrases that Suppes interprets as sets can be represented as concepts. And those he interprets as binary relations can be represented as roles. Take for example the phrase *male vegetarians*. According to Figure 3 its algebraic representation is $[male] \cap [vegetarian]$ which according to Figure 1 translates to (and male vegetarian) with male and vegetarian denoting concepts respectively representing the set of males and the set of vegetarians.

As subset relations correspond to subsumption relations and Peirce product corresponds to a some term, a terminological representation of the sentence *John loves Mary* as interpreted in (9) is

(16) John $\sqsubseteq$ (some love Mary).

Recall that proper nouns are mapped to singleton sets. Accordingly, John and Mary denote concepts interpreted as singleton sets. As an aside, (9) can also be represented as an ABox statement. Namely:

(17) (assert-ind John Mary love).

This representation is equivalent to the terminological representation in (16). Here, John and Mary denote ABox elements, that is, elements of concepts. In general, an assertional statement of the form (assert-ind a b R) is interpreted as $(a, b) \in R$, where $a, b \in U$ are the interpretations of the ABox elements a and b.

Terminological formulations for the quantified verb phrases in (10) are:

(18)	Verb phrase	Terminological representation
(i)	*eat all fruit*	(not (some (not eat) fruit))
(ii)	*eat (some) fruit*	(some eat fruit)
(iii)	*eat no fruit*	(not (some eat fruit))

Observe that the algebraic interpretation (10) (i) of verb phrases quantified with *all* is not the variant (4) of Peirce product that is associated with the all operator. For representing verb phrase of this form we need a representation language that provides for roles to be negated. With the exception of the languages $\mathcal{U}$ and $\mathcal{KL}$ most terminological languages (including $\mathcal{ALC}$, BACK, CLASSIC and $\mathcal{KRIS}$) don't.

The all construct is useful for representing verb phrases such as *eat only fruit*. According to the semantics given in (13) a linguistically adequate terminological representation is the conjunction:

(19) (and (some eat fruit) (all eat fruit)).

Quantified sentences like those in (11) can be formulated as subsumption, equivalence and disjointness relations on concepts. The terminological representations for (11) are:

(20)	Sentence	Terminological representation
(i)	*Some persons eat (some) fruit*	(and persons (some eat fruit)) $\neq\ \bot$
(ii)	*All persons eat (some) fruit*	persons $\sqsubseteq$ (some eat fruit)
(iii)	*No person eats (some) fruit*	(disjoint persons (some eat fruit))

The inequality in (i) is strictly speaking not a well-formed terminological definition. However, for any concept C, C $\neq \bot$ is semantically equivalent to

(21) (some ∇ C) $\doteq \top$.

This follows as for any set C the following is true: $C \neq \emptyset$ iff $U^2 : C = U$. Hence we may use the inequality of (20) (i) as an abbreviation for

(22) (some ∇ (and persons (some eat fruit))) $\doteq \top$.

The work of Suppes and Böttner can also be utilised to provide valuable assistance for the reverse translation process from given terminological expressions (formulated with those operators that have algebraic associations), into their English formulations. For example, given the terminological statement

(23) (disjoint boys (not (some love girls)))

its algebraic representation is

(24) $[boys] \cap ([love] : [girls])' \ = \ \emptyset$

which translates to *No boy loves no girls* according to Suppes' denoting grammar. Note that there are algebraic representations without corresponding natural language formulations. Examples are algebraic representations of the form $[noun]'$ and $[verb] : [noun]'$. Thus, not every terminological expression has a English translation in the fragment analysed by Suppes and Böttner.

Nevertheless, I believe the work of Suppes and Böttner provides a useful link between natural language and terminological representation. Their work provides a formal basis for simplifying the translation process between representational expressions and natural language. It shows the extent to which English formulations can be expressed with the set of terminological operators listed in Section 2. And it contributes to a better understanding of the different terminological operators. For example, from a linguistic point of view the all construct is often used incorrectly. Here is a typical example from the literature. In Patel-Schneider (1990, p. 14) the term

(25) (and person (all child lawyer))

is said to define 'the class of people whose children are all lawyers'. This description is ambiguous. People whose children are all lawyers could refer to people who are parents of all lawyers, that is, people who for each person who is a lawyer are parents of that person. Or it could refer to people who are parents only of lawyers. The intended meaning is the latter. But according to Böttner, the representation (25) is not adequate. See the discussion on the semantics of *eat only fruit* in Section 3 according to which a linguistically adequate representation of the set of 'people who are parents only of lawyers' is

(26) (and person (some parent lawyer) (all parent lawyer)).

Note that the assumed reading of the role child in (25) is different from the reading I assume in this paper. In (25) child represents the relation 'has as child' (and not 'is a child of') whereas in (26) parent represents 'is a parent of'.

I conclude with some loose observations. First, I believe the translation processes from natural language statements of the kind accommodated in the algebraic context to terminological representations could be automated. I envisage an implementation with three components. Given some natural language expression one component computes the syntactic structure (in the form of a syntactic derivation tree, for example) in accordance with a phrase-structure grammar. The second component then constructs the semantic representation (in the form of a semantic tree, for example) thus deriving the algebraic representation. And in the third component the algebraic representation is transformed into a terminological representation. Of course, as natural language is ambiguous the derived terminological representations needn't be unique. Negated verb phrases, for example, have more than one possible representation. The reverse process of generating the English language formulations for terminological representations would proceed in the opposite direction. Note however, as not every terminological expression has a natural English language formulation this process will be incomplete.

Second, the fragments of the English language representable in the representation languages $\mathcal{U}$ and $\mathcal{KL}$ are supersets of that fragment representable in the algebraic language. In $\mathcal{U}$ and $\mathcal{KL}$ we can also define *number restrictions* of the form *John loves at least* 3 *girls*, *John loves at most* 2 *girls* and *John loves exactly* 1 *girl*. These have no relation-algebraic representations.

Third, in (1981) Suppes proposes a natural deduction calculus for 'direct inference in English'. Whether terminological reasoners exist that may be used for this purpose requires further investigation. Systems with expressive terminological languages like $\mathcal{U}$ and $\mathcal{KL}$ are candidates. However, to my knowledge neither $\mathcal{U}$ nor $\mathcal{KL}$ are implemented in a knowledge representation system. Inference in such a system would be undecidable. Schild (1988) showed that there is no algorithm for deciding whether a subsumption relationship of $\mathcal{U}$ is true.

Fourth, the work of Suppes and Böttner may also be relevant in other areas besides KL-ONE-based knowledge representation. There is a link to the work of McAllester and Givan (1989) and Givan, McAllester and Shalaby (1991) who (similar to Suppes (1981)) aim at the development of a formalism for direct inference in natural language. Their representation language is related to the language of Montague (1974) and provides separate operators for quantification of the kind in (10) (i).

Finally, I want to stress that not every linguistic phenomenon can be characterised in the algebraic framework. This was not Suppes' intention. His intention was to analyse the extent to which natural language can be accomodated in the context of (extended) relation algebra. As there is a direct link between relation algebra and terminological representation Suppes' and Böttner's investigations also cast some light on the extent to which natural language can be accomodated in the context of terminological representation formalisms.

5 Acknowledgements

I would like to thank the following people for many useful comments: Michael Böttner, Chris Brink, Bernhard Hollunder, Franz Baader, Ullrich Hustadt, Hans Jürgen Ohlbach and Barbara Zimmermann. For financial support I thank the Deutsche Forschungsgemeinschaft, SFB 314 (TICS).

References

Baader, F. and Hollunder, B. (1991), $\mathcal{KRIS}$: $\mathcal{K}$nowledge $\mathcal{R}$epresentation and $\mathcal{I}$nference $\mathcal{S}$ystem: System description, *Technical Memo TM-90-03*, Deutsches Forschungszentrum für Künstliche Intelligenz, Kaiserslautern, Germany.

Borgida, A., Brachman, R. J., McGuinness, D. L. and Resnick, L. A. (1989), CLASSIC: A structural data model for objects, *Proceedings of the ACM SIGMOD-89 International Conference on Management of Data*, Portland, Oreg., pp. 58–67.

Böttner, M. (1985), Variable-free semantics for 'only' and 'except', Manuscript, Institut für Kommunikationsforschung und Phonetik, Universität Bonn, Germany.

Böttner, M. (1989), Variable-free semantics for anaphora, Manuscript, Department of Mathematics, University of Cape Town, South Africa. To appear in *Journal of Philosophical Logic*.

Böttner, M. (1990). Personal communication.

Böttner, M. (1991), Boolean modules and extended relation algebras, Manuscript, Institute for Mathematical Studies in the Social Sciences, Stanford University, Stanford, California.

Böttner, M. (1992), State transition semantics, *Theoretical Linguistics* **18**(2/3), 239–286.

Brachman, R. J. (1990), The future of knowledge representation, *Proceedings of the National Conference on Artificial Intelligence (AAAI)*, pp. 1082–1092. Extended Abstract.

Brachman, R. J. and Schmolze, J. G. (1985), An overview of the KL-ONE knowledge representation system, *Cognitive Science* **9**(2), 171–216.

Brachman, R. J., Gilbert, V. P. and Levesque, H. J. (1985), An essential hybrid reasoning system: Knowledge and symbol level accounts of KRYPTON, *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI)*, Los Angeles, California, pp. 532–539.

Brink, C. (1981), Boolean modules, *Journal of Algebra* **71**(2), 291–313.

Brink, C. and Schmidt, R. A. (1992), Subsumption computed algebraically, *Computers and Mathematics with Applications* **23**(2–9), 329–342. Special Issue on Semantic Networks in Artificial Intelligence.

Brink, C., Britz, K. and Schmidt, R. A. (1992), Peirce algebras, *Technical Report MPI-I-92-229*, Max-Planck-Institut für Informatik, Saarbrücken, Germany.

Donini, F. M., Lenzerini, M., Nardi, D. and Nutt, W. (1991), The complexity of concept languages, *in* J. Allen, R. Fikes and E. Sandewall (eds), *Proceedings of the Second International Conference on Principles of Knowledge Representation and Reasoning*, Morgan Kaufmann, San Mateo, CA, pp. 151–162.

Doyle, J. and Patil, R. S. (1991), Two theses of knowledge representation: Language restrictions, taxonomic classification, and the utility of representation services, *AI* **48**, 261–297.

Givan, R., McAllester, D. and Shalaby, S. (1991), Natural language based inference procedures applied to Schubert's steamroller, *Proceedings of the National Conference on Artificial Intelligence (AAAI)*, pp. 915–920.

Hayes, P. (1977), In defense of logic, *International Joint Conference on Artificial Intelligence* pp. 559–565.

Jónsson, B. (1982), Varieties of relation algebras, *Algebra Universalis* **15**(3), 273–298.

Levesque, H. J. and Brachman, R. J. (1987), Expressiveness and tractability in knowledge representation and reasoning, *Computational Intelligence* **3**, 78–93.

MacGregor, R. M. (1991), Inside the LOOM description classifier, *SIGART Bulletin* **2**(3), 88–92. Special Issue on Implemented Knowledge Representation and Reasoning Systems.

Maddux, R. D. (1991a), Introductory course on relation algebras, finite-dimensional cylindric algebras, and their interconnections, *in* H. Andréka, J. D. Monk and I. Németi (eds), *Algebraic Logic*, Vol. 54 of *Colloq. Math. Soc. J. Bolyai*, North-Holland, Amsterdam, pp. 361–392.

Maddux, R. D. (1991b), The origin of relation algebras in the development and axiomatization of the calculus of relations, *Studia Logica* **50**(3/4), 421–455.

McAllester, D. and Givan, R. (1989), Natural language syntax and first order inference, *Memo 1176*, MIT Artificial Intelligence Laboratory. To appear in AIJ.

McDermott, D. (1978), Tarskian semantics, or no notation without denotation!, *Cognitive Science* **2**(3), 277–282.

Montague, R. (1974), English as a formal language, *in* R. H. Thomason (ed.), *Formal Philosophy: Selected Papers of Richard Montague*, Yale University Press, pp. 188–221.

Nebel, B. (1988), Computational complexity of terminological reasoning in BACK, *Artificial Intelligence* **34**, 371–383.

Nebel, B. and von Luck, K. (1988), Hybrid reasoning in BACK, *in* Z. W. Ras and L. Saitta (eds), *Methodologies for Intelligent Systems*, Vol. 3, NH, pp. 260–269.

Patel-Schneider, P. F. (1987), *Decidable, Logic-Based Knowledge Representation*, PhD thesis, University of Toronto.

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

Patel-Schneider, P. F. (1990), Practical, object-based knowledge representation for knowledge-based systems, *Information Systems* **15**(1), 9–19.

Schild, K. (1988), Undecidability of subsumption in $\mathcal{U}$, *KIT-Report 67*, Department of Computer Science, Technische Universität Berlin, Germany.

Schmidt, R. A. (1991), *Algebraic terminological representation*, M.Sc. thesis, Thesis-Reprint TR 011, Department of Mathematics, University of Cape Town, Cape Town, South Africa. Also available as Technical Report MPI-I-91-216, Max-Planck-Institut für Informatik, Saarbrücken, Germany.

Schmidt-Schauß, M. (1989), Subsumption in KL-ONE is undecidable, *in* R. J. Brachman, H. J. Levesque and R. Reiter (eds), *Proceedings of the First International

Conference on Principles of Knowledge Representation and Reasoning, Morgan Kaufmann, San Mateo, CA, pp. 421–431.

Schmidt-Schauß, M. and Smolka, G. (1991), Attributive concept description with complements, *Artificial Intelligence* **48**, 1–26.

Schmolze, J. G. and Mark, W. S. (1991), The NIKL experience, *Computational Intelligence* **6**, 48–69.

Suppes, P. (1973), Semantics of context-free fragments of natural languages, *in* K. J. J. Hintikka, J. M. E. Moravcsik and P. Suppes (eds), *Approaches to Natural Language*, Reidel Publ. Co., Dordrecht, Holland, pp. 370–394.

Suppes, P. (1976), Elimination of quantifiers in the semantics of natural language by use of extended relation algebras, *Rev. Int. de Philosophie* **30**(3–4), 243–259.

Suppes, P. (1979), Variable-free semantics for negations with prosodic variation, *in* E. Saarinen, R. Hilpinen, I. Niiniluoto and M. P. Hintikka (eds), *Essays in Honour of Jaakko Hintikka*, Reidel Publ. Co., Dordrecht, Holland, pp. 49–59.

Suppes, P. (1981), Direct inference in English, *Teaching Philosophy* **4**, 405–418.

Tarski, A. (1941), On the calculus of relations, *Journal of Symbolic Logic* **6**(3), 73–89.

Woods, W. A. (1975), What's in a link: Foundations for semantic networks, *in* Bobrow and Collins (eds), *Representation and Understanding: Studies in Cognitive Science*, Academic Press, pp. 35–82.

Woods, W. A. and Schmolze, J. G. (1992), The KL-ONE family, *Computers and Mathematics with Applications* **23**(2–5), 133–177. Special Issue on Semantic Networks in Artificial Intelligence.