Relational Grammars for Knowledge Representation

Renate A. Schmidt

Centre for Agent Research and Developments
Department of Computing and Mathematics
Manchester Metropolitan University, Chester Street,
Manchester M1 5GD, United Kingdom

Abstract

This paper aims to enhance the practical applicability of relational grammars, which have been devised for the semantic analysis of natural language. We focus on their application in knowledge representation. In particular, we address how the representation problem for KL-ONE-based knowledge representation systems can be automatically solved with the help of relational grammars. New rules are presented for natural language formulations, like *has sons* and *has at least two sons*, commonly arising in application domains. For accommodating the latter kind of sentences we introduce a new class of so-called (concrete) graded Peirce algebras. A graded Peirce algebra is a Peirce algebra endowed with a countable set of numerical quantifier operations.

1 Introduction

Relational grammars were introduced by P. Suppes (1973, 1976), arising from a dissatisfaction with first-order logic as an instrument appropriate for the semantic analysis of natural language. Instead he proposes an algebraic approach, using so-called extended relation algebras. Extended relational algebras are set-theoretic structures of particular set-relation interactions which are captured in the abstract framework of Peirce algebras (Brink, Britz and Schmidt 1994).

Also in the seventies, R. J. Brachman (1977, 1979) had the idea to amalgamate the useful features of semantic networks and frames in a system called KL-ONE. This was the starting point for the area of KL-ONE-based knowledge representation. The initial goal was to develop notational devices for representing the meaning of any concepts expressible in natural language. The main emphasis of subsequent research has been on logical and algorithmic aspects, especially the decidability of a number of subsystems of KL-ONE. Research had focussed less on natural language specific issues. In (1990) Brachman says:

'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.'

There have been a variety of applications involving natural language processing (for example, Quantz and Schmitz 1994, Fehrer et al. 1994, Franconi 1994). But, as far as I know, there has not been a formal and general attempt to formalise the connection between knowledge representation and natural language. As first argued in Schmidt (1993) I believe relational grammars achieve exactly this. This is based on the observation that many of subsystems of KL-ONE are in essence just multi-modal logics or the logical versions of (reducts of) Peirce algebras (Schild 1991, Brink and Schmidt 1992, Brink et al. 1994, de Rijke 1994).

Relational grammars in their current form have two deficiency, though, which this paper addresses.

(i) Relational grammars do not accommodate numerical information as in *a city is a place with more than 100 000 inhabitants.*

(ii) Sentences, like *Elizabeth has sons*, cannot be dealt with.

The aim of this paper is twofold. One, we show that relational grammars provide a formal scheme for solving the representation problem of knowledge representation. The representation problem is concerned with translating natural language descriptions of information into the language of a given KL-ONE-type system, so that the information can be stored in the database and can be manipulated by the inference mechanism. Moreover, we illustrate how the translation from natural language to KL-ONE-type languages can be automated.

Two, we remove the deficiencies (i) and (ii). This is done by enhancing Peirce algebras with the algebraic counterparts of numerical quantifier operations available in KL-ONE-based systems and graded modal logics. Incorporating numerical quantifiers in relational grammars solves some of the problems noted in Böttner (1994a, 1994b).

The paper is organised as follows. Section 2 defines the algebraic structures common to computational linguistics, knowledge representation and modal logic, in particular, Peirce algebras and modal algebras. Section 3 describes relational grammars and their use for natural language analysis. Section 4 is devoted to KL-ONE-type knowledge representation. Section 5 describes how we can make use of relational grammars for automatically finding adequate KL-ONE representations for information formulated in English. Section 6 discusses numerical quantification from knowledge representation and graded modal logic, and introduces concrete graded Peirce algebras. These are utilised in an extension of relational grammars described in Section 7. The final section is the Conclusion.

2 Extended relation algebras, Peirce algebras and modal algebras

This section describes algebras of sets and relations interacting with each other, in particular, extended relation algebras, Peirce algebras and their modal reducts, and their interrelationships.

An extended relation algebra is in essence a mixture of a Boolean set algebra and an algebra of binary relations. Formally, an *extended relation algebra* $\mathcal{E}(U)$ over a non-empty set U, 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.[1] 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 A is the set

$$R \text{``} A = \{y \mid \exists x \ (x,y) \in R \text{ and } x \in A\}.$$

The *domain restriction* of a relation R to a set A is given by

$$R \restriction A = \{(x,y) \mid (x,y) \in R \text{ and } x \in A\}.$$

The algebraic counterparts of extended relation algebras are Peirce algebras introduced in Brink et al. (1994). A *Peirce algebra* is a two-sorted algebra $(\mathfrak{B}, \mathfrak{R}, :, {}^c)$ of a Boolean algebra $\mathfrak{B} = (B, +\cdot, ', 0, 1)$ and a relation algebra $\mathfrak{R}$ that interact via the *Peirce product* $:$ of Boolean modules and a cylindrification operation ^{c}.

An algebra $\mathfrak{R} = (R, +, \cdot, ', 0, 1, ;, {}^\smile, e)$ is a *relation algebra* if $(R, +, \cdot, ', 0, 1)$ is a Boolean algebra, $;$ is an associative operation with identity e, ${}^\smile$ is an involution, both $;$ and ${}^\smile$ distribute over $+$, ${}^\smile$ distributes as follows over $;$

$$(r\,; s)^\smile = s^\smile ; r^\smile,$$

and the following holds

$$r^\smile ; (r\,; s)' \ \leq \ s',$$

for any $r, s \in R$ (Tarski 1941). The standard model of an abstract relation algebra is a concrete relation algebra, which is any subalgebra of the algebra $\mathbf{2}^{U \times U}$ of all binary relations. In concrete relation algebras $;$ is relational composition, ${}^\smile$ is converse and the identity element is the identity relation, that is,

$$\begin{aligned} R\,; S \ &= \{(x,y) \mid \exists z \ (x,z) \in R \text{ and } (z,y) \in S\} \\ R^\smile \ &= \{(x,y) \mid (y,x) \in R\}. \end{aligned}$$

The combination $(\mathfrak{B}, \mathfrak{R}, :)$ of a Boolean algebra and a relation algebra by the Peirce product $:$ is a *Boolean module* (Brink 1981), provided $:$ is a mapping

[1] This definition extends the original definition by Suppes (1976) with domain restriction.

from $\mathfrak{R} \times \mathfrak{B}$ into $\mathfrak{B}$ with the following properties. The Peirce product distributes over addition, that is, for any $r, s \in R$ and $a, b \in B$,

$$r:(a+b) = r:a + r:b \quad \text{and} \quad (r+s):a = r:a + s:a,$$

a kind of weak associativity holds,

$$r:(s:a) = (r\,;s):a,$$

e and 0 are the identity and the zero of $:$, and

$$r^{\smile}:(r:a)' \;\leq\; a'.$$

Adding a further operation we obtain a *Peirce algebra*. An algebra $(\mathfrak{B}, \mathfrak{R}, :, ^c)$ is a Peirce algebra provided $(\mathfrak{B}, \mathfrak{R}, :)$ is a Boolean module and ^{c} is an operation from $\mathfrak{B}$ to $\mathfrak{R}$ such that for every $a \in B$ and $r \in R$,

$$a^c:1 = a \quad \text{and} \quad (r:1)^c = r\,;1.$$

Set-theoretically the Peirce product is a multiplication of a relation R and a set A bearing the set

$$R:A = \{x \mid \exists y\ (x,y) \in R \text{ and } x \in A\}.$$

The cylindrification operation maps a set A to the relation

$$A^c = \{(x,y) \mid x \in A\}.$$

Extended relations algebras provide set-theoretic models for Peirce algebras, since both image and domain restriction can be expressed with Peirce product and the cylindrification operation (for details see Brink et al. 1994). To capture exactly the class of concrete Peirce algebras $(\mathfrak{B}(U), \mathfrak{R}(U), :, ^c)$ with $\mathfrak{B}(U)$ being the powerset algebra over U and $\mathfrak{R}(U)$ being the full algebra of all relations over U, we require further conditions. It can be shown that any complete and atomic Peirce algebra in which the relational atoms satisfy $1\,;p\,;1 = 1$ and $p^{\smile}\,;1\,;p \;\leq\; e$, is isomorphic to an algebra $(\mathfrak{B}(U), \mathfrak{R}(U), :, ^c)$. This can be proved by mimicking the representation theorem of full relation algebras. A modal characterisation of full Peirce algebras is by de Rijke (1998).

An advantage of relation algebras and Peirce algebras is their natural equational axiomatisation. But, from a practical point of view a disadvantage is that both the elementary and the equational theory of relation algebras is only semi-decidable. The same holds for Peirce algebras. Decidability is an important criterion for a large fraction of the KL-ONE-community. The most popular KL-ONE-systems can be accommodated in decidable reducts of Peirce algebras, in particular, in modal algebras, which we will define next. Modal algebras are obtained by removing the relational operations from Peirce algebras. Relational composition and relational negation are known to be responsible for the undecidability of the equational theory of relation algebras (Kurucz et al. 1993).

A *modal algebra* is a Boolean algebra with a family of unary operators $(\mathfrak{B}, \{\Diamond_r \mid r \in I\})$.[2] Each operator is normal and additive, that is,

$$\Diamond_r(0) = 0 \quad \text{and} \quad \Diamond_r(a+b) = \Diamond_r(a) + \Diamond_r(b).$$

The Peirce product has exactly these properties, so that for any Boolean module $(\mathfrak{B}, \mathfrak{R}, :)$ or Peirce algebra $(\mathfrak{B}, \mathfrak{R}, :, ^c)$ the reduct $(\mathfrak{B}, \{\Diamond_r \mid r \in I\})$ with $\Diamond_r$ defined by $\Diamond_r(a) = r : a$, for any $a \in B$, is a modal algebra. Set-theoretically,

$$\Diamond_R(A) = R : A$$

defines the semantics of a multi-modal expression of the form $\Diamond_\alpha \varphi$. Recall, $\Diamond_\alpha \varphi$ is true in a state x iff there is a state y accessible from x via the transition α, interpreted by R, in which φ is true. The dual operator is $\Box_R$ and is defined by

(1) $$\Box_R(A) = (R : A')' = \{x \mid \forall y \ (x, y) \in R \text{ implies } x \in A\}.$$

Modal algebras are expressively weaker than Peirce algebras and have some pleasant properties. The class of modal algebras is representable (which follows by specialisation from Jónsson and Tarski 1951 & 1952). The equational theory is decidable.

In contrast to knowledge representation, in computational linguistics decidability is not an essential prerequisite, on the contrary, some may argue, compared to first-order logic, for example, the algebraic framework is expressively too weak for serious linguistic analysis. Indeed, there are constructs in relational grammars of Böttner (1994b) requiring more structure than just sets and binary relations. More expressive power can be achieved by incorporating, for example, projection functions into the algebraic framework (Maddux 1997).

3 Relational grammars

Relational grammars define the semantics of a fragment of English by annotating syntactic grammars with algebraic expressions. Suppes (1976) originally refers to relational grammars as potentially denoting grammars. We will sketch a variation of the definition of relational grammars by example.[3] Formal descriptions can be found in the work of Suppes (1976, 1979, 1981) and Böttner (1992a, 1992b, 1994a, 1994b, 1995, 1997b). A survey paper is Böttner (1997a).

Let U denote the domain of interpretation. U is a non-empty set. The denotation of sentences and phrases is given by a mapping $[\cdot]$ from syntactic types into the concrete Peirce algebra $(\mathfrak{B}(U), \mathfrak{R}(U), :, ^c)$ over U (or the equivalent extended relation algebra). It describes a subset of the hierarchy of sets and binary relations $\mathbf{2}^U \cap \mathbf{2}^{U^2}$ which is closed under the algebraic operations. $[\cdot]$

[2]This definition of a modal algebra is a natural generalisation of the definition given by Jónsson (1993).

[3]We do not define the Fregean function for the annotation of the rule S $\longrightarrow$ PN + VP in Figure 2.

$$\begin{aligned}
\mathrm{S} &\longrightarrow \mathrm{PN} + \mathrm{VP} \\
\mathrm{VP} &\longrightarrow \mathrm{TV} + \mathrm{EQ} + \mathrm{NP} \\
\mathrm{NP} &\longrightarrow \mathrm{Adj} + \mathrm{N}
\end{aligned}$$

Figure 1: A phrase structure grammar

$$\begin{array}{rll}
\mathrm{S} & \longrightarrow \mathrm{PN} + \mathrm{VP} & [\mathrm{PN}] \subseteq [\mathrm{VP}] \\
\mathrm{VP} & \longrightarrow \mathrm{TV} + \mathrm{EQ} + \mathrm{NP} & [\mathrm{TV}] : [\mathrm{NP}] \\
\mathrm{NP} & \longrightarrow \mathrm{Adj} + \mathrm{N} & [\mathrm{Adj}] \cap [\mathrm{N}]
\end{array}$$

Figure 2: A relational grammar

is defined inductively by a valuation on elementary types and algebraic operations that determine the denotation of non-elementary types. This will be made precise in the following.

The denotation of the *elementary types* is defined by a valuation mapping v into subsets of U or binary relations over U. Elementary types are, for example, nouns, adjectives, intransitive verbs which v maps into the Boolean set algebra $\mathfrak{B}(U)$. Proper nouns are special elementary types as they are mapped to singleton sets, that is, atoms in the Boolean set algebra. Other elementary types are transitive verbs, and they are mapped into the relation algebra $\mathfrak{R}(U)$. The pair (U, v) is called a *model structure*.

The denotations of *non-elementary* phrases and sentences are then defined, in so-called *relational grammars*, by algebraic combinations from the denotations of the elementary types. A relational grammar is a phrase structure grammar extended with the semantic function $[\cdot]$. Recall, a *phrase structure grammar* defines the syntax of a language and is specified in terms of a set of production rules like those of Figure 1. The symbols S, NP, VP, TV, EQ, Adj, N and PN denote 'start symbol', 'noun phrase', 'verb phrase', 'transitive verb', 'existential quantifier', 'adjective', 'noun' and 'proper noun', respectively. For the fragment of English which Suppes and Böttner study the phrase structure grammar is context-free. The extension with semantic annotations for the syntactic grammar of Figure 1 is the relational grammar of Figure 2. It is merely a small extract of the tables presented in the papers of Suppes and Böttner. Each production rule has an algebraic annotation defining the semantics for the relevant types.

The following examples illustrate how, in the context of relational grammars, the algebraic operations combine the denotations of the elementary types in oder to form denotations for non-elementary types.

The operations of the Boolean sort $\mathfrak{B}(U)$ are union, intersection and complement. The following are sample phrases that are interpreted inside the Boolean set algebra.

$$
\begin{aligned}
[\text{males and females}] &= [\text{males}] \cup [\text{females}] \\
[\text{both princes and heirs}] &= [\text{princes}] \cap [\text{heirs}] \\
[\text{adult princes}] &= [\text{adult}] \cap [\text{princes}] \\
[\text{not heirs}] &= [\text{heirs}]'
\end{aligned}
$$

Inside relational grammars the denotation of the noun phrase *males and females* is defined to be the union of the denotations of the nouns *males* and *females*. Nouns are elementary types and their denotations are given by the valuation mapping v. It maps nouns to sets. Hence, [males] $= v(\textit{males})$ and [females] $= v(\textit{females})$ are both in $\mathbf{2}^U$. Relational grammars define the denotation of the phrase *both princes and heirs* by the intersection of the denotation of the nouns *princes* and *heirs*, and the denotation of and the adjective-noun phrase *adult princes* by the intersection of the denotation of its adjective and its noun. The interpretation of negative verb phrases like *are not heirs* is the complement of the denotation of the noun *heirs*. In each case the value of the function $[\cdot]$ on the right hand sides of the above identities is determined by v.

The Peirce product, the multi-modal diamond operator, combines sets with relations. By this operation the denotations of relative primitives, for example, transitive verbs, can be combined with the denotations of nouns to form the denotation of a verb phrase. For example, the phrases *are fathers of princes*, *are fathers of some princes* or *is a father of (some) prince* are all given the following denotation.

$$[\text{father(s) of (some) princes}] = [\text{father}] : [\text{princes}]$$

There are two dual variants of the Peirce product, the involution and backward involution operators, which capture two kinds of universal quantifications (Peirce 1870). The backward involution operator corresponds to the modal box operator and was defined above, in (1). From a linguistic perspective, this operation is sometimes used incorrectly in the knowledge representation literature. The asymmetry in the meaning of phrases involving the universal quantifier word *all* and the exclusive *only* is often overlooked. This is illustrated by the next two examples interpreted in accordance with Böttner (1994b). The phrase *princes all of whose sons are princes* is interpreted by the intersection of the set of princes and a box expression:

$$[\text{princes all of whose sons are princes}] = [\text{princes}] \cap ([\text{son}]^{\smile} : [\text{princes}]')'.$$

Compare this interpretation with the interpretation of the phrase *are fathers of princes only*:

(2) $$[\text{fathers of princes only}] = [\text{father}] : [\text{princes}] \cap ([\text{father}] : [\text{princes}]')'.$$

It involves a diamond expression, in addition to the box expression, as the box part has no existential import (in general, it does not hold that $(R : A')'$ is a subset of $R:A$). The diamond part [father] : [princes] is necessary to ensure that the semantics adequately captures our intuition that a father of princes only is also a father of some prince.

The involution operator is defined by

$$(R':A)' = \{x \mid \forall y\ x \in A \rightarrow (x,y) \in R\}\,.$$

It is a universal quantifier operation of the kind suitable for interpreting phrases like *are fathers of every prince.*

$$[\text{fathers of every princes}] = ([\text{father}]' : [\text{princes}])'$$

Transitive verbs are among the elementary types of the language fragment of Suppes and Böttner that are interpreted as binary relations. Operations by which relations can be combined to form new relations are those of $\mathfrak{R}(U)$. These are the Boolean operations of union, intersection and complementation, as well as composition and converse. Here are examples of the algebraic semantics for a negated transitive verb and transitive verbs combined by a conjunctive.

$$\begin{aligned} [\text{is not father of}] &= [\text{father}]' \\ [\text{is both father and teacher of}] &= [\text{father}] \cap [\text{teacher}] \end{aligned}$$

The relational composition chains together two transitive verbs, and the converse operation is useful for capturing the denotation of passives:

$$\begin{aligned} [\text{are mothers of (some) parents of}] &= [\text{mother}] \,;\, [\text{parent}] \\ [\text{is loved by}] &= [\text{loves}]^{\smile}. \end{aligned}$$

Examples of phrases captured by relation-forming operations like domain restriction or Cartesian product (which, by the way, is also inter-definable with the cylindrification operation in Peirce algebra) are:

$$\begin{aligned} [\text{male parent of}] &= [\text{parent}] \upharpoonright [\text{male}] \\ [\text{Charles and Andrew are brothers}] &= [\text{Charles}] \times [\text{Andrew}] \ \subseteq\ [\text{brother}]. \end{aligned}$$

4 Knowledge representation

This section reviews knowledge representation of the kind concerned with KL-ONE-based systems. Comprehensive recent survey papers are Donini et al. (1996) and Woods and Schmolze (1992).

Knowledge bases of KL-ONE-type systems usually consist of two components, the TBox and the ABox. The TBox is the 'terminological' part of the knowledge base expressing relationships between concepts and roles (or sets and relations), whereas the ABox is very much like a database and contains 'assertional' information about individuals. Before we make this precise we give two examples of typical entries in a TBox and an ABox, which are chosen from practical applications. In an application of describing and classifying multimedia (Goble, Haul and Bechhofer 1996), a video which covers male politicians is specified by

$$\begin{aligned}&\mathsf{Video\text{-}which\text{-}covers\text{-}male\text{-}politicians} \doteq \\ &\quad \mathsf{Video} \sqcap \exists\, \mathsf{cover} \,.\, (\mathsf{person} \sqcap \exists\, \mathsf{hasSex} \,.\, \mathsf{male} \sqcap \exists\, \mathsf{worksFor} \,.\, \mathsf{PoliticalParty}).\end{aligned}$$

A medical record entry in an application of building and maintaining a terminology server for a medical domain (Rector et al. 1997) is specified by

$$(\exists\, \mathsf{hasFracture} \,.\, (\exists\, \mathsf{hasLocation} \,.\, \mathsf{Femur}))(\mathsf{Mrs\ Smith}).$$

This says, Mrs Smith has a fracture which is located in the femur.

The TBox part is formalised by so-called *terminological* or *description logics*, of which there are a great variety. We will briefly define a prototypical description logic with numerical quantifier operations, namely, a variation of the logic $\mathcal{ALCN}$ (Hollunder and Baader 1991), which is called $\mathcal{ALCN}^+$ (Ohlbach, Schmidt and Hustadt 1996). $\mathcal{ALCN}^+$ is an extension with numerical quantifier operations of the popular description logic, called $\mathcal{ALC}$ (Schmidt-Schauß and Smolka 1991).

Formally, the language of any description logic consists of a set of primitive roles and a disjoint set of primitive concepts. Two designated primitive concepts are the top concept $\top$ and the bottom concept $\bot$. From primitive roles R and primitive concepts A compound concept terms C, D, ... are formed according to the following rule.

$$\mathsf{C} \quad \longrightarrow \quad \mathsf{A} \mid \neg\mathsf{C} \mid \mathsf{C} \sqcap \mathsf{C} \mid \mathsf{C} \sqcup \mathsf{C} \mid \exists\, \mathsf{R} \,.\, \mathsf{C} \mid \forall\, \mathsf{R} \,.\, \mathsf{C} \mid \exists_{\geq n}\, \mathsf{R} \,.\, \mathsf{C} \mid \exists_{\leq n}\, \mathsf{R} \,.\, \mathsf{C}$$

n denotes a non-negative integer. Relationships between concept descriptions are expressed by TBox statements, or *terminological statements*, which have one of two forms:

$$\mathsf{C} \sqsubseteq \mathsf{D} \quad \text{or} \quad \mathsf{C} \doteq \mathsf{D}.$$

A TBox is defined to be a finite set of terminological statements.

The semantics of $\mathcal{ALCN}^+$ is specified by an interpretation $(U, \cdot^{\mathcal{I}})$ with U a non-empty set (the domain of interpretation) and a valuation mapping $\cdot^{\mathcal{I}}$. The valuation maps roles to binary relations over U and it maps concepts to subsets of U with the following restrictions.

$$\begin{aligned}\top^{\mathcal{I}} &= U \\ \bot^{\mathcal{I}} &= \emptyset \\ (\neg\mathsf{C})^{\mathcal{I}} &= U \setminus \mathsf{C}^{\mathcal{I}} = (\mathsf{C}^{\mathcal{I}})' \\ (\mathsf{C} \sqcap \mathsf{D})^{\mathcal{I}} &= \mathsf{C}^{\mathcal{I}} \cap \mathsf{D}^{\mathcal{I}} \\ (\mathsf{C} \sqcup \mathsf{D})^{\mathcal{I}} &= \mathsf{C}^{\mathcal{I}} \cup \mathsf{D}^{\mathcal{I}}\end{aligned}$$

$$\begin{aligned}\exists\, \mathsf{R} \,.\, \mathsf{C}^{\mathcal{I}} &= \mathsf{R}^{\mathcal{I}} : \mathsf{C}^{\mathcal{I}} \\ \forall\, \mathsf{R} \,.\, \mathsf{C}^{\mathcal{I}} &= (\mathsf{R}^{\mathcal{I}} : (\mathsf{C}^{\mathcal{I}})')' \\ \exists_{\geq n}\, \mathsf{R} \,.\, \mathsf{C}^{\mathcal{I}} &= \{x \mid |\{y \in \mathsf{C}^{\mathcal{I}} \mid \mathsf{R}^{\mathcal{I}}(x,y)\}| \geq n\} \\ \exists_{\leq n}\, \mathsf{R} \,.\, \mathsf{C}^{\mathcal{I}} &= \{x \mid |\{y \in \mathsf{C}^{\mathcal{I}} \mid \mathsf{R}^{\mathcal{I}}(x,y)\}| \leq n\}\end{aligned}$$

A terminological statement $\mathsf{C} \sqsubseteq \mathsf{D}$, respectively $\mathsf{C} \doteq \mathsf{D}$, is true in an interpretation $(U, \cdot^{\mathcal{I}})$ iff $\mathsf{C}^{\mathcal{I}} \subseteq \mathsf{D}^{\mathcal{I}}$, respectively $\mathsf{C}^{\mathcal{I}} = \mathsf{D}^{\mathcal{I}}$. More generally, an interpretation is a model of a TBox T iff every statement in T is true in the interpretation. A concept C is said to be *subsumed* by another concept D with respect to T iff $\mathsf{C} \sqsubseteq \mathsf{D}$ is true in every model of T.

The ABox component of a knowledge base captures information about individual elements, in concepts and roles of the TBox, by a finite set of statements of the form

$$\mathsf{C}(\mathsf{a}) \quad \text{or} \quad \mathsf{R}(\mathsf{a},\mathsf{b}).$$

The symbols a and b denote individual names. The semantics is defined inside the interpretations of the TBox component, by: $\mathsf{a}^{\mathcal{I}}, \mathsf{b}^{\mathcal{I}} \in U$, and $\mathsf{C}(\mathsf{a})$ and $\mathsf{R}(\mathsf{a},\mathsf{b})$ are true in a interpretation $(U, \cdot^{\mathcal{I}})$ iff

$$\mathsf{a}^{\mathcal{I}} \in \mathsf{C}^{\mathcal{I}} \quad \text{and} \quad (\mathsf{a}^{\mathcal{I}}, \mathsf{b}^{\mathcal{I}}) \in \mathsf{R}^{\mathcal{I}},$$

respectively. An individual a is an *instance* of a concept C with respect to a TBox T iff $\mathsf{a}^{\mathcal{I}} \in \mathsf{C}^{\mathcal{I}}$ in all models of T. Similarly, a pair (a, b) is an instance of R with respect to T iff $(\mathsf{a}^{\mathcal{I}}, \mathsf{b}^{\mathcal{I}}) \in \mathsf{R}^{\mathcal{I}}$ in every model of T.

Subsumption checking and the instance problem are two typical inference problems of KL-ONE-based systems. Another typical problem is concerned with the classification of concepts in the TBox. This means, computing subsumption relationships between any distinct pair of concepts (or roles in description logics more expressive than $\mathcal{ALC}$ and $\mathcal{ALCN}^+$).

Evidently, subsumption and instance checking in the combination of an ABox and a TBox defined by $\mathcal{ALC}$ can be formalised in the context of modal algebras, when the ABox individuals are interpreted as singleton sets and ABox statements as subset relationships. More expressive description logics include also operations for forming compound roles (for example conjunctions, disjunctions, composition, etcetera) and for expressing relationships between roles. Most of these can be accommodated inside atomic and simple Peirce algebras (Brink et al. 1994, Brink and Schmidt 1992). In fact, most of the additional operations not available in $\mathcal{ALC}$ or $\mathcal{ALCN}$ are just notational variants of the operations of relation algebras, cylindrification, forms of residuation plus other constructs inter-definable in terms of the basic operations of Peirce algebra. This correspondence is exploited in the subsequent sections.

There is also a connection of KL-ONE-based systems to modal logic (Schild 1991, de Rijke 1994, van der Hoek and de Rijke 1995). For instance, $\mathcal{ALC}$ corresponds exactly to the multi-modal version of the basic modal logic K, and $\mathcal{ALCN}^+$ to graded modal logic. The latter connection will be utilised in Section 6.

5 The representation problem

This section sketches by way of an example how relational grammars can be utilised for solving the representation problem of knowledge representation, as

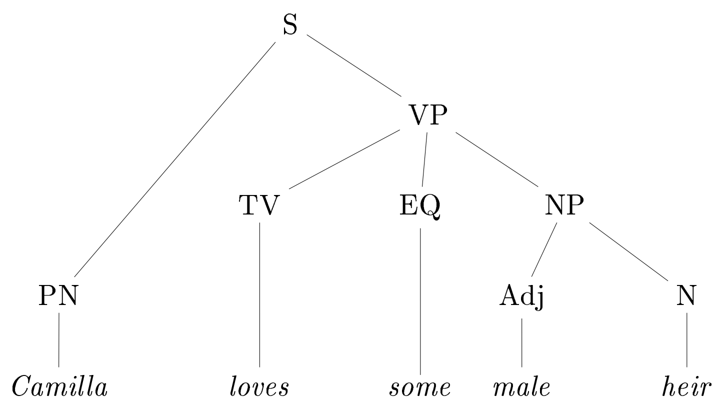

Figure 3: A sample derivation tree

first proposed in Schmidt (1993).

The representation problem is the problem of finding adequate KL-ONE-representations for given information formulated in ordinary English. For example, we are given the information that

Camilla loves some male heir

and aim to find a linguistically adequate representation. The syntactic structure of the sample sentence as defined by the phrase structure grammar of Figure 1 is represented by the syntactic *derivation tree* of Figure 3. Its semantics is defined in two parts. One part is the relational grammar, namely that of Figure 2, and the other part is the model structure (U, v) that defines the denotations for the elementary types. For our example, $v : \{\mathrm{PN}, \mathrm{Adj}, \mathrm{N}\} \longrightarrow \mathbf{2}^U$ and $v : \{\mathrm{TV}\} \longrightarrow \mathbf{2}^{U^2}$.

According to the relational grammar the semantic association for adjective-noun phrases is the intersection of the denotation of the types adjective and noun. Adjectives and nouns are elementary types whose denotation is defined by the valuation function v. The algebraic interpretation of the phrase *male heirs* is the intersection of the denotations of *male* and *heirs*. This can be read off at the node NP of the Figure 4. It depicts an annotated derivation tree referred to as a *semantic tree*. The procedure for constructing the semantic tree is the following:

(i) Start at the leaves of the syntactic derivation tree,

(ii) use the mapping v to assign meaning of the terminal symbols that have elementary types, and then

(iii) use the table of the relational grammar and systematically derive the semantics for the entire sentence.

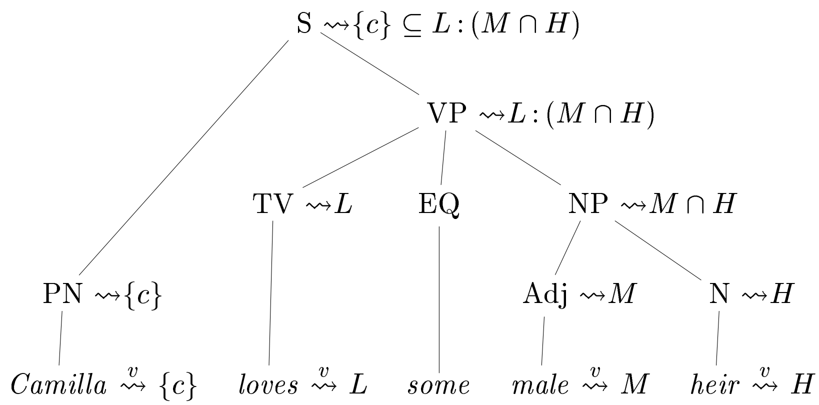

Figure 4: A semantic tree

v is a partial function on the terminal symbols of the grammar. For example, the quantifier word *some* is not of type elementary and has no denotation. Its meaning is reflected by the Peirce product of the denotation of *loves* and the denotation of *male heirs*. The label of the root symbol, that is,

$$\{c\} \subseteq L : (M \cap H), \tag{3}$$

defines the semantics of the entire sentence.

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 and the model structure of the elementary types.

Finding the KL-ONE-representation for the sample sentence is now straightforward, given the exact correspondence between the algebraic operations and the KL-ONE constructs explained in the previous section. The reformulation of (3) in KL-ONE syntax is:

$$(\exists\, \mathsf{loves} \,.\, (\mathsf{male} \sqcap \mathsf{heirs}))(\mathsf{Camilla}).$$

The example illustrates two points. One, natural language analysis in the context of relational algebras provides a formal analysis of KL-ONE-like languages. Relational grammars give some insight into the extent to which English formulations can be expressed with the vocabulary of KL-ONE and contribute to a better understanding of the expressiveness of the different KL-ONE-languages.

Two, relational grammars provide a formal framework for systematically translating information formulated in a natural language so that the information can be stored and manipulated inside a knowledge representation system. Relational grammars couple syntax and semantics completely, and this tight fit, we believe, makes automating the translation process possible. In some rare cases going the reverse direction, translating KL-ONE expressions or algebraic expressions into natural language, is also possible. However, not every KL-ONE or algebraic expression has a natural English language formulation. An example is the expression $\forall\,\mathsf{father\text{-}of}\,.\,\mathsf{princes}$. As discussed earlier in Section 3, for it to represent the concept of *fathers of princes only* a condition saying fathers of princes exist, is missing. However, the concept $\forall\,\mathsf{has\text{-}son}\,.\,\mathsf{princes}$ has a natural language description, it is the concept of *all those, all of whose sons are princes.* The crucial difference is, whether the relative is interpreted actively or passively (in terms of the converse operation).

The same observations are true when we use relational grammars to transform natural language into formulae of multi-modal logics, like propositional modal logic and dynamic modal logic.

6 Numerical quantification

Numerical quantifier operations are standard operations in many KL-ONE-based systems (extending $\mathcal{ALCN}$), but have not been used in the context of relational grammars, and they are not available in either relation algebras or Peirce algebras, not even implicitly. In this section we will introduce numerical quantifier operations to relational grammars and Peirce algebras. We will do so by exploiting the link to $\mathcal{ALCN}^+$ and graded modal logics, in particular. The literature on graded modal logic is extensive, references can be found in van der Hoek and de Rijke (1995) and Ohlbach et al. (1996). In the setting of algebraic logic counting quantifiers are treated in Mikulás (1995) and Andréka, Hodkinson and Németi (1996).

The numerical quantifier operations of graded modal logics and KL-ONE-based logics can be used to express the existence of n objects as in the sentences:

(4)
Elizabeth owns more than three castles
Charles is a father of at least two boys
Charles has at most one sister
Charles has exactly three adult siblings
Elizabeth has less than five children
More than 50 million subjects of Elizabeth are British.

This KL-ONE statement

$$\mathsf{city} \doteq \mathsf{place} \sqcap (\mathsf{atleast}\ 100\,001\ \mathsf{inhabited\text{-}by}\ \mathsf{people})$$

defines a city as a place inhabited by at least 100 001 people. It is impossible to formulate statements like this one in the framework of Peirce algebras. With regards to definability the elementary theory of unary and binary relations is a fragment of first-order logic with no more than three different variables,[4] but a first-order formulation of 'there are at least n ... ' requires $n+1$ different variables.

[4] With regards to derivability a fourth variable is required (see Maddux 1997).

The modal equivalent formulations for the numerical quantifier expressions $\exists_{\geq n}\,\mathsf{R}\,.\,\mathsf{C}$ and $\exists_{\leq n}\,\mathsf{R}\,.\,\mathsf{C}$ are

$$\blacklozenge_{R,n-1}\varphi \quad \text{and} \quad \blacksquare_{R,n}\neg\varphi,$$

respectively. $\blacklozenge_{R,n}$ and $\blacksquare_{R,n}$ are so-called *graded modalities.*

$\blacklozenge_{R,n}\varphi$ is read to mean φ is true in *more than* n worlds accessible by R, and
$\blacksquare_{R,n}\varphi$ is read to mean $\neg\varphi$ is true in *at most* n worlds accessible by R.

Based on the axiomatisation of graded modal logic (found in van der Hoek and de Rijke 1995) we will now define their algebraic counterparts.

A *graded modal algebra* is an algebra

$$(\mathfrak{B}, \{f_{r,0} \mid r \in I\}, \{f_{r,n} \mid r \in I, n \in \mathbb{N}\}),$$

where I is any non-empty set such that $(\mathfrak{B}, \{f_{r,0} \mid r \in I\})$ is a modal algebra and each $f_{r,n}$ with $n > 0$ is a mapping $\mathfrak{B} \longrightarrow \mathfrak{B}$ governed by the following quasi-equations. We write $r :_n a$ for $f_{r,n}(a)$. For any $a, b \in B$ and any $n \geq 0$:

G1 $r :_{n+1} a \;\leq\; r :_n a$
G2 $(r :_0 (a \cdot b))' \cdot r :_n a \;\leq\; r :_n b'$
G3 $(r :_0 (a \cdot b))' \cdot r :_{!n} a \cdot r :_{!m} b \;\leq\; r :_{!n+m} (a + b)$

where $r :_{!0} a = (r :_0 a)'$ and $r :_{!n} a = r :_{n-1} a \cdot (r :_n a)'$ for $n > 0$.

It is not difficult to prove the operations $f_{r,n}$ with $n > 0$ are normal (that is, $r :_n 0 = 0$), and monotone (that is, if $a \leq b$ then $r :_n a \leq r :_n b$). However, unlike $f_{r,0}$, they are not additive. The inclusion $r :_n a + r :_n b \;\leq\; r :_n (a + b)$ holds, but not the reverse inclusion.

The intended semantics of an (abstract) graded modal algebra is given by the notion of a *concrete graded modal algebra.* We define a concrete graded modal algebra to be a subalgebra of the *full concrete graded modal algebra* over a non-empty set U and a subset $\{R_i\}_i$ of binary relations on U:

(5) $$(\mathfrak{B}(U), \{f_{R_i,0}\}_i, \{f_{R_i,n} \mid n \in \mathbb{N}\}_i),$$

where $\mathfrak{B}(U)$ is a power set algebra over U, and each $f_{R,0}$ and $f_{R,n}$ is a mapping $\mathbf{2}^U \longrightarrow \mathbf{2}^U$ defined by

(6) $$f_{R,n}(A) = R :_n A = \{x \mid |A \cap R\,``\{x\}| > n\}$$

for $n \geq 0$ and $A \in \mathbf{2}^U$. The dual operation is:

$$(R :_n A')' = \{x \mid |A' \cap R\,``\{x\}| \leq n\}.$$

In words, $R :_n A$ is the set of elements which have more than n relatives by R in A, and $(R :_n A')'$ is the set of elements which have at most n relatives by R in the complement of A. As expected, $f_{R,0}(A) = R :_0 A$ coincides with the Peirce product $R : A$.

We will not address the issue of the representability of graded modal algebras. As an aside we note, it is easy to verify that any concrete graded modal algebra is an abstract graded modal algebra. And, any frame $\mathcal{F} = (U, \{R_i\}_i)$ determines a graded modal algebra. Based on the soundness and completeness of graded modal logic, we expect the converse holds also, namely, any graded modal algebra determines a concrete graded modal algebra.

In the next section we will make use of concrete Peirce algebras endowed with a countable set of counting quantifiers, defined by subalgebras of

$$(\mathfrak{B}(U), \mathfrak{R}(U), :, {}^c, \{ :_n \mid n \in \mathbb{N}\})$$

with $:_n$ given by (6). These algebras will be referred to as *concrete graded Peirce algebras*. The new operations satisfy the axioms G1, G2 and G3. We leave open, whether there is also an axiomatisation of the interactions with the extra-modal operations.

7 Enhancing relational grammars

This section introduces new rules to relational grammars in order to accommodate formulations common in knowledge representation applications. We will focus on sentences, like *Elizabeth has adult sons*, and sentences with counting expressions, like (4), which have not been treated before inside relational grammars.

Verb phrases involving *has* or *have*

Our proposed rules interpret constellations with *has* (respectively *have*) differently in different contexts. For example, the meaning of the following sentences will be as given in the right column:

Elizabeth has sons	$[\text{Elizabeth}] \in [\text{son}]^{\smile} : U$
Elizabeth has castles	$[\text{Elizabeth}] \in [\text{has}] : [\text{castles}]$.

Reformulated in first-order notation the respective interpretations are

$$\exists x\ (x, E) \in S \quad \text{and} \quad \exists x\ (E, x) \in H \wedge x \in C.$$

The word *has* is accommodated differently depending on whether it occurs together with a relational noun or not. In the first sentence it is not assigned a denotation. This distinction is made by the rules:

$$\text{VP} \longrightarrow \textit{has/have} + \text{RNP} \qquad\qquad [\text{RNP}]^{\smile} : U$$

for the combination with a relational noun phrase and the standard rule, otherwise:

$$\text{VP} \longrightarrow \text{TV} + \text{NP} \qquad\qquad [\text{TV}] : [\text{NP}].$$

RNP is the syntactic type of relational noun phrases and is determined by the rules

$$\text{RNP} \longrightarrow \text{RN} \qquad [\text{RN}]$$

$$(7) \qquad \text{RNP} \longrightarrow \text{Adj} + \text{RN} \qquad [\text{RN}] \downharpoonleft [\text{Adj}],$$

where RN is the type of relational nouns (for example, *son*, *sister*, *sibling*, *child*, *subject*) and $\downharpoonleft$ is range restriction, given by $R \downharpoonleft A = (R^{\smile} \upharpoonright A)^{\smile}$. Thus, the denotation of *Elizabeth has adult sons* is derived in the familiar way, by invoking the first rule of Figure 2. Accordingly, its denotation is

$$[\text{Elizabeth}] \in ([\text{son}] \downharpoonleft [\text{adult}])^{\smile} : U,$$

or equivalently, $[\text{Elizabeth}] \in ([\text{son}]^{\smile} \upharpoonright [\text{adult}]) : U$.

Other forms of combinations of *has* with relational noun phrases are

Elizabeth has some sons

Elizabeth has a son.

Their denotations are defined as specified by:

$$\text{VP} \longrightarrow \textit{has/have} + \text{EQ} + \text{RNP} \qquad [\text{RNP}]^{\smile} : U$$

$$\text{VP} \longrightarrow \textit{has/have} + \text{IArt} + \text{RNP} \qquad [\text{RNP}]^{\smile} : U$$

with IArt being the type of indefinite articles.

Two open problems come to mind. The rule (7) does not apply to intensive adjectives, like *young* or *small*, which require special treatment (Böttner 1994b). So one open problem is, how can the semantics of *Elizabeth has young grandsons* be captured? Another open problem is, how can exclusive information, as in *Charles has sons only*, be accommodated. It seems here background assumptions are needed, namely, someone who has sons only is someone who has children and all the children are male, but 'children' or 'male' are not explicit in the sentence.

Numerical quantification

In the semantic analysis of natural language numbers are usually interpreted as sets governed by Peano's axioms. For example, the number 2 is interpreted to be the set of all sets containing exactly two elements. A possible denotation of the phrase *two sons* is then $[2] \cap \mathbf{2}^{[\text{son}]}$. This solution is not feasible in relational grammars (Böttner 1994a, 1994b). We propose an alternative solution by which the meaning of counting expressions is formulated in terms of the operations of concrete graded Peirce algebras.

For this purpose we extend the definition a relational grammar (given in Section 3). There is a new syntactic type, namely the cardinality type Card, which will be mapped to non-negative integers. The hierarchy of sets and binary relations is restricted by a concrete graded Peirce algebra $(\mathfrak{B}(U), \mathfrak{R}(U), :, ^c, \{ :_n | n \in$

$\mathbb{N}\}$) over a non-empty set U (introduced at the end of the previous section). The meaning function $[\cdot]$ is now a mapping from syntactic types to the disjoint union of the non-negative integers and the concrete graded Peirce algebra. In particular, $[\cdot]$ maps the cardinality type Card to numbers and the remaining types in the familiar way to elements of the algebra. The remainder of this section assumes

$$[\mathrm{Card}] = n.$$

In the new enhanced algebraic context, denotations of sentences and phrases involving cardinalities can be accommodated by the rules we will define now. The semantics of phrases, like

owns more than three castles
owns at least three castles
owns at most three castles
owns less than three castles
owns exactly three castles
owns three castles,

derive with the rules

$$\begin{array}{lll}
 & \mathrm{VP} \longrightarrow \mathrm{TV} + \textit{more than} + \mathrm{Card} + \mathrm{NP} & [\mathrm{TV}] :_n [\mathrm{NP}] \\
 & \mathrm{VP} \longrightarrow \mathrm{TV} + \textit{at least} + \mathrm{Card} + \mathrm{NP} & [\mathrm{TV}] :_{n-1} [\mathrm{NP}] \\
 & \mathrm{VP} \longrightarrow \mathrm{TV} + \textit{at most} + \mathrm{Card} + \mathrm{NP} & ([\mathrm{TV}] :_n [\mathrm{NP}]')' \\
(8) & \mathrm{VP} \longrightarrow \mathrm{TV} + \textit{less than} + \mathrm{Card} + \mathrm{NP} & ([\mathrm{TV}] :_{n-1} [\mathrm{NP}]')' \\
 & \mathrm{VP} \longrightarrow \mathrm{TV} + \textit{exactly} + \mathrm{Card} + \mathrm{NP} & [\mathrm{TV}] :_{!n} [\mathrm{NP}] \\
 & \mathrm{VP} \longrightarrow \mathrm{TV} + \mathrm{Card} + \mathrm{NP} & [\mathrm{TV}] :_{n-1} [\mathrm{NP}].
\end{array}$$

Thus, *owns three castles* is interpreted like *owns at least three castles.* The semantics of phrases, like

inhabited by more than 100 000 *people*,

and the variations with *at least*, *at most*, etcetera, derive from the respective variations of the rule

$$\mathrm{VP} \longrightarrow \mathrm{TV} + \textit{ed} + \textit{by} + \textit{more than} + \mathrm{Card} + \mathrm{NP} \qquad [\mathrm{TV}] :_n [\mathrm{NP}].$$

The rules for *fathers of more than two princes* and its variations are similar. The sample sentences *Charles has at most one sister*, *Charles has exactly three adult siblings* and *Elizabeth has less than five children* are treated according to the scheme determined by the set of rules in (8) and the following:

$$\mathrm{VP} \longrightarrow \textit{has} + \textit{more than} + \mathrm{Card} + \mathrm{RNP} \qquad [\mathrm{RNP}]^{\smile} :_n U.$$

Numerical quantification can occur also in the subject position. The semantics of the sentences

At least two princes like books
At least two young princes like books,

and the like, is reflected by the rules

S $\longrightarrow$ More than + Card + NP + VP	$U^2 :_n ([\mathrm{NP}] \cap [\mathrm{VP}]) = U$
S $\longrightarrow$ At least + Card + NP + VP	$U^2 :_{n-1} ([\mathrm{NP}] \cap [\mathrm{VP}]) = U$
S $\longrightarrow$ At most + Card + NP + VP	$(U^2 :_n ([\mathrm{NP}] \cap [\mathrm{VP}])')' = U$
S $\longrightarrow$ Less than + Card + NP + VP	$(U^2 :_{n-1} ([\mathrm{NP}] \cap [\mathrm{VP}])')' = U$
S $\longrightarrow$ Exactly + Card + NP + VP	$U^2 :_n ([\mathrm{NP}] \cap [\mathrm{VP}]) = U$
S $\longrightarrow$ Card + NP + VP	$U^2 :_{n-1} ([\mathrm{NP}] \cap [\mathrm{VP}]) = U.$

This is consistent with the treatment of the sentence *at least* n *women are young* in a modal context by van der Hoek and de Rijke (1995). The modal interpretation is $\blacklozenge_n(W \wedge Y)$ where $\blacklozenge_n$ is an *S5*-operator (which is defined by the universal relation). Remember the correspondence between modal logics and algebras is given by the equivalence: φ is valid in a model (U, R, v) iff $v(\varphi) = U$, where v is the extended valuation function.

The sentences

Some princes like books and *At least one prince likes books*

are different formulations of the same information. In relational grammars they transform to different, but equivalent, expressions:

$$[\text{princes}] \cap [\text{like books}] \neq \emptyset \quad \text{and} \quad U^2 :_0 ([\text{princes}] \cap [\text{like books}]) = U.$$

(since $A \neq \emptyset$ iff $U^2 : A = U$, and $U^2 : A = U^2 :_0 A$).

Relational counting expressions in the subject position, as in

At least two children like books
At least two children who are princes like books
More than 50 *million subjects of Elizabeth are British*,

are captured by rules similar to

$$\text{S} \longrightarrow \text{More than + Card + SRNP + VP} \qquad U^2 :_n ([\mathrm{SRNP}] \cap [\mathrm{VP}]) = U.$$

The semantics of the relational noun phrase type SRNP (the S indicates the phrase occurs in subject position) is given by:

SRNP $\longrightarrow$ RN	$[\mathrm{RN}] : U$
SRNP $\longrightarrow$ Adj + RN	$[\mathrm{Adj}] \cap [\mathrm{RN}] : U$
SRNP $\longrightarrow$ RN + RelPr + Cop + NP	$[\mathrm{NP}] \cap [\mathrm{RN}] : U$
SRNP $\longrightarrow$ RN + *of* + PN	$[\mathrm{PN}] \cap [\mathrm{RN}] : U.$

RelPr is short for relative pronoun (for example, *who*, *which*, *that*) and Cop for copula (for example, *is*, *are*).

8 Conclusion

In summary, this paper discusses how natural language can be translated automatically via relational grammars into KL-ONE-like knowledge representation languages. It enhances relational grammars in order to treat natural language specifications common in knowledge representation applications. In particular, we introduce designated rules for combinations of *has* and *have* with relational noun phrases. And, we define concrete graded Peirce algebras in order to accommodate counting expressions.

Acknowledgements

I thank especially Michael Böttner for his help, Ullrich Hustadt and Miroslava Tzakova for their comments, and Maarten Marx for pointers to the literature. This work was completed at the Max-Planck-Institut für Informatik, Saarbrücken, Germany, and was financially supported by the DFG project TRA-LOS.

References

Andréka, H., Hodkinson, I. and Németi, I. (1996), Finite algebras of relations are representable on finite sets. To appear in *Journal of Symbolic Logic*.

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

Böttner, M. (1992b), Variable-free semantics for anaphora, *Journal of Philosophical Logic* **21**, 375–390.

Böttner, M. (1994a), Open problems in relational grammars, *Patrick Suppes: Scientific Philosopher*, Vol. 3, Kluwer, pp. 19–39.

Böttner, M. (1994b), Relationale Sprachen. Habilitationsschrift, Max-Planck-Institut für Psycholinguistik, Nijmegen.

Böttner, M. (1995), A collective extension of relational grammar. Paper presented at RelMiCS II, Parati, Brasil. To appear in *Bull. IGPL*.

Böttner, M. (1997a), Natural language, *in* C. Brink, W. Kahl and G. Schmidt (eds), *Relational Methods in Computer Science*, Advances in Computing, Springer, Wien, pp. 226–246.

Böttner, M. (1997b), Visiting some relatives of Peirce's, *Proc. RelMiCS III*, Hammamet, Tunisia, pp. 311–320.

Brachman, R. J. (1977), What's in a concept: Structural foundations for semantic networks, *Int. J. Man-Machine Studies* **9**, 127–152.

Brachman, R. J. (1979), On the epistomological status of semantic networks, *in* N. V. Findler (ed.), *Associative Networks: Representation and Use of Knowledge by Computers*, Academic Press, pp. 3–50.

Brachman, R. J. (1990), The future of knowledge representation, *Proc. AAAI'90*, pp. 1082–1092.

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

Brink, C., Britz, K. and Schmidt, R. A. (1994), Peirce algebras, *Formal Aspects of Computing* **6**(3), 339–358.

Brink, C. and Schmidt, R. A. (1992), Subsumption computed algebraically, *Computers and Mathematics with Applications* **23**(2–5), 329–342.

de Rijke, M. (1994), Meeting some neighbours, *in* J. van Eijck and A. Visser (eds), *Logic and Information Flow*, MIT Press, pp. 170–195.

de Rijke, M. (1998), A modal characterization of Peirce algebras, *in* E. Orlowska (ed.), *Logic at Work*, Springer, pp. 109–123.

Donini, F. M., Lenzerini, M., Nardi, D. and Schaerf, A. (1996), Reasoning in description logics, *in* G. Brewka (ed.), *Principles in Knowledge Representation*, Studies in Logic, Language and Information, CSLI Publications, Stanford, pp. 191–236.

Fehrer, D., Hustadt, U., Jaeger, M., Nonnengart, A., Ohlbach, H. J., Schmidt, R. A., Weidenbach, C. and Weydert, E. (1994), Description logics for natural language processing, In *Proc. International Workshop on Description Logics '94*, Document D-94-10, DFKI, Saarbrücken, Germany, pp. 80–84.

Franconi, E. (1994), Description logics for natural language processing. In *Working Notes of the AAAI'94 Fall Symposium: KR for NLP in Implemented Systems*, New Orleans.

Goble, C. A., Haul, C. and Bechhofer, S. (1996), Describing and classifying multimedia using the description logic GRAIL, *Proc. IS&T/SPIE Conf. on Storage and Retrieval of Still Image and Video IV*, San Jose.

Hollunder, B. and Baader, F. (1991), Qualifying number restrictions in concept languages, *in* J. Allen, R. Fikes and E. Sandewall (eds), *Proc. KR'91*, Morgan Kaufmann, San Mateo, pp. 335–346.

Jónsson, B. (1993), A survey of Boolean algebras with operators, *in* I. G. Rosenberg and G. Sabidussi (eds), *Algebras and orders*, Vol. 389 of *NATO Adv. Sci. Inst. Ser. C Math. Phys. Sci.*, Kluwer, Dordrecht, pp. 239–286.

Jónsson, B. and Tarski, A. (1951 & 1952), Boolean algebras with operators, Part I & II, *American Journal of Mathematics* **73 & 74**, 891–939 & 127–162.

Kurucz, Á., Németi, I., Sain, I. and Simon, A. (1993), Undecidable varieties of semilattice-ordered semigroups, of Boolean algebras with operators and logics extending Lambek calculus, *Bull. IGPL* **1**(1), 91–98.

Maddux, R. D. (1997), Relation algebras, *in* C. Brink, W. Kahl and G. Schmidt (eds), *Relational Methods in Computer Science*, Advances in Computing, Springer, Wien, pp. 22–38.

Mikulás, S. (1995), *Taming Logics*, PhD thesis, University of Amsterdam.

Ohlbach, H. J., Schmidt, R. A. and Hustadt, U. (1996), Translating graded modalities into predicate logic, *in* H. Wansing (ed.), *Proof Theory of Modal Logic*, Vol. 2 of *Applied Logic Series*, Kluwer, pp. 253–291.

Peirce, C. S. (1870), Description of a notation for the logic of relatives, resulting from an amplification of the conceptions of Boole's calculus of logic, *Mem. Amer. Acad.* **9**, 317–378.

Quantz, J. J. and Schmitz, B. (1994), Knowledge-based disambiguation for machine translation, *Minds and Machines* **4**, 39–57.

Rector, A., Bechhofer, S., Goble, C. A., Horrocks, I., Nowlan, W. A. and Solomon, W. D. (1997), The GRAIL concept modelling language for medical terminology, *Artificial Intelligence in Medicine* **9**, 139–171.

Schild, K. (1991), A correspondence theory for terminological logics: Preliminary report, *Proc. IJCAI'91*, pp. 466–471.

Schmidt, R. A. (1993), Terminological representation, natural language & relation algebra, *in* H. J. Ohlbach (ed.), *Proc. GWAI'92*, Vol. 671 of *Lecture Notes in Artificial Intelligence*, Springer, pp. 357–371.

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

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, Dordrecht, 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, Dordrecht, 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.

van der Hoek, W. and de Rijke, M. (1995), Counting objects, *Journal of Logic and Computation* **5**(3), 325–345.

Woods, W. A. and Schmolze, J. G. (1992), The KL-ONE family, *Computers and Mathematics with Applications* **23**(2–5), 133–177.