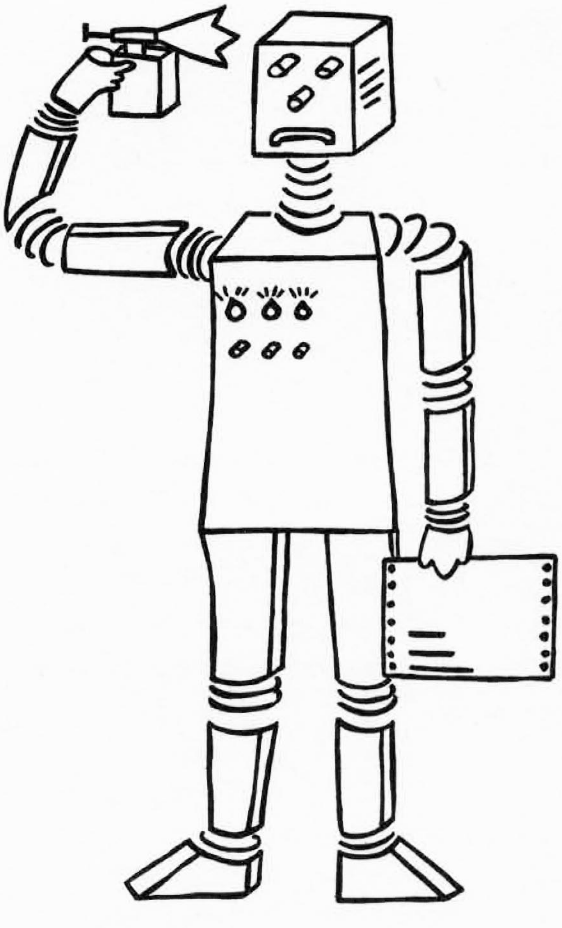
SEKI-REPORT

Artificial
Intelligence
Laboratories

Fachbereich Informatik
Universität Kaiserslautern
Postfach 3049
D-6750 Kaiserslautern 1, W. Germany



# Subsumption in KL-One is Undecidable

Manfred Schmidt-Schauß

SEKI Report SR-88-14

# Subsumption in KL-ONE is Undecidable

Manfred Schmidt-Schauß*

## Abstract.

It is shown that in the frame-based language KL-ONE it is undecidable, whether one concept is subsumed by another concept. The concept forming operators which are sufficient for this result are: conjunction, value restriction, role restriction, the operator 'SOME', and role value maps using only '='. In particular, number restrictions are not used. This shows that there is a basic difference between feature terms and KL-ONE, since the complexity of subsumption switches from co-NP-complete to undecidable, if the restriction is dropped that roles are functional.

Key words: Knowledge representation, classification, subsumption, KL-ONE

* Fachbereich Informatik, Universität Kaiserlautern, Po-box 3049, D-6750 Kaiserslautern, F.R. Germany

# 1. Introduction.

The frame-based knowledge representation language KL-ONE [BS85, BGL85] permits describing concepts on the basis of unary predicates (concepts, frames) and binary predicates (roles, slots). KL-ONE or related languages are currently being used as the basis of several knowledge representation systems [BS85, LN87, KBR86, Mg88], in particular for natural language processing.

The terminological component (T-Box) of KL-ONE gives a user the possibility to structure a domain of interest by using concepts and roles (frames and slots). Usually, such a description starts by postulating certain primitive concepts and roles, and afterwards describing definable concepts using the operators available. New concepts can be defined via conjunction, via value restriction with respect to some role ("a person with every male friend is a doctor"), via number restriction (" a person with more than three children"), and via role value maps ("a man with: every child of a child of his father is also a child").

Research in computational linguistics has lead to a related knowledge representation method on the basis of unification grammars [Sh86], namely so-called $\Psi$-terms [AN86] or feature terms [SA87, Sm88], where concepts are called sorts and roles are called features. The difference to knowledge representation á la KL-ONE is that feature terms permit only functional roles. The term forming rules for feature terms directly correspond to the concept forming possibilites of KL-ONE. Indeed, the expressiveness of feature terms can be viewed as KL-ONE where all roles are functional, and where a complement operator is added.

The basic service provided by all these languages is a reasoning facility called 'classification' that informs the user whether a concept A is subsumed by a concept B, that means: every B is also an A. For several languages related to KL-ONE, the complexity of subsumption is known: The language $\mathcal{FL}^-$ admits a classification algorithm of quadratic time complexity [LB87], the language $\mathcal{FL}$ has a co-NP-complete classification problem. Recently it was shown that the classification problem for feature terms is co-NP-complete [Sm88].

Although there is agreement upon that role value maps provide complications, there are systems allowing them as descriptive possibility and use them also in the (incomplete) classification algorithm. In this paper it is shown that subsumption in a sublanguage of KL-ONE is undecidable. Number restrictions are used in this proof only in the restricted form of a 'some' operator, wheras role value maps are crucial for this proof. However role value maps are only permitted with '=' as comparison operator.

This result was a surprise, since the very similar language of feature terms has a decidable (co-NP-complete) classification problem. This means that if we extend the feature term language by allowing arbitrary roles instead of functions then the complexity of classification switches from co-NP-complete to undecidable. This shows that role value maps in the restricted form are acceptable, but that the general form as used in KL-ONE should be avoided.

Recently, P. Patel-Schneider [PS88] has shown that classification in KL-ONE is undecidable. However, the used sublanguage of KL-ONE appears to require more of the expressivity of KL-ONE than our sublanguage, such as role value maps with '$\subseteq$' as comparison operator, inverse roles and number restrictions. A further proof of undecidable subsumption of roles in a KL-ONE related language including negation of roles was recently presented by K. Schild [Sc88].

## 2. The Language

In the following we describe the syntax and semantics of an extension of the terminological language KL-ONE [BS85] in a slightly modified syntax, but the same semantics. Our language KL-ONE$_{+C}$ allows to construct concept expressions as follows. There are disjoint sets of role symbols and concept symbols. Concept expressions are:

i) concept symbols
ii) $C \sqcap D$, if C and D are concept expressions.
iii) $\forall R{:}C$, if R is a role symbol and C is a concept expression.
iv) $\exists R{:}C$, if R is a role symbol and C is a concept expression.
v) $P = Q$ if P and Q are lists of roles.
vi) $\neg C$ if C is a concept expression.

Let R,S be relations over a set M, i.e., $R,S \subseteq M \times M$.

The composition of R and S is defined as:

$R \circ S := \{(x,y) \mid \exists z \in M\ (x,z) \in R \wedge (z,y) \in S\}$.

The application of a relation R to a set $s \subseteq M$ is defined as follows:

$sR := \{y \mid \exists x{:}\ x \in s \wedge (x,y) \in R\}$.

The application of R to an element is defined analogously:

$xR := \{y \mid (x,y) \in R\}$.

Obviously, we have $s\,(R \circ S) = (sR)S$

The semantics of the concept expressions is as usual. An **interpretation** $I$ is a pair (M, I), where M is a set and I an interpretation function, such that

i) for every concept symbol C: $I(C) \subseteq M$
ii) for every role symbol R: $I(R) \subseteq M \times M$.

Lists of roles are interpreted as the composition of relations:

$I((R_1,\ldots,R_n)) = I(R_1) \circ \ldots \circ I(R_n)$.

We intepret defined concepts as subsets of M as follows:

$$
\begin{aligned}
I(C \sqcap D) &= I(C) \cap I(D)\\
I(C \sqcup D) &= I(C) \cup I(D)\\
I(\forall R{:}C) &= \{x \in M \mid \forall y{:}\ (x,y) \in I(R) \Rightarrow y \in I(C)\}\\
I(\exists R{:}C) &= \{x \in M \mid \exists y{:}\ (x,y) \in I(R) \wedge y \in I(C)\}.\\
I(P = Q) &= \{x \in M \mid x(I(P)) = x(I(Q))\}\\
I(\neg C) &= M - I(C).
\end{aligned}
$$

Subsumption and Consistency are defined with respect to this semantics:

A concept expression C **subsumes** a concept expression D, iff for all possible interpretations $I$, we have $I(C) \supseteq I(D)$.

A concept C is **consistent**, iff there exists an interpretation I, such that $I(C) \neq \emptyset$, otherwise it is called **inconsistent.**

Two concepts C and D are **equivalent**, iff C subsumes D and D subsumes C.

**2.1 Lemma.** Let C, D be concept expressions.

Then D subsumes C, iff $\neg D \sqcap C$ is an inconsistent concept. ■

Since we have negation in $\text{KL-ONE}_{+C}$, subsumption problems are equivalent to inconsistency problems.

There is a minor difference to usual KL-ONE: The operator $\exists R{:}C$ appears to introduce more expressivity. However, this operator can be seen as an abbreviation of (SOME (RESTRICT R C)), hence $\text{KL-ONE}_{+C}$ without complements is a sublanguage of KL-ONE. The sublanguage of $\text{KL-ONE}_{+C}$ without complements is called KL-ONE*. Obviously this is a sublanguage of KL-ONE. In KL-ONE* it is not possible to define new roles, and number restrictions appear only in the form of the SOME. Furthermore role value maps of the form $(P \subseteq Q)$ are disallowed.

## 3. Subsumption is Undecidable.

Similar as in [SS88, Sm88], we transform subsumption problems into a system $\mathcal{C}$ of constraints, where every single constraint is of one of the forms $s \subseteq C$, $x \in C$, $s \neq t$. We write x,y, z for element variables, s,t denote expressions of the form $xR_1 \ldots R_n$, and C, D denote concept expressions.

Let $I = (M, I)$ be an interpretation. Let $\alpha$ be an assignment of elements in M to element variables of $\mathcal{C}$. We assume that $\alpha$ extends the interpretation function I, i.e.$\alpha C = I(C)$, $\alpha(xP) := (\alpha x)I(P)$.

Then we say $\alpha$ **satisfies** $\mathcal{C}$, if the following holds:

for $(x \in A) \in \mathcal{C}$, we have $\alpha x \in \alpha A$

for $(A \subseteq B) \in \mathcal{C}$, we have $\alpha(A) \subseteq \alpha B$

for $(A = B) \in \mathcal{C}$, we have $\alpha A = \alpha B$.

for $(A \neq B) \in \mathcal{C}$, we have $\alpha A \neq \alpha B$.

A constraint system $\mathcal{C}$ is **consistent**, iff there exists a model M and an assignment $\alpha$ with respect to this model such that $\alpha$ satisfies $\mathcal{C}$, otherwise it is called **inconsistent**.

**3.1 Lemma.** Let C be a concept expression.

Then C is consistent, iff the constraint system $\{x \in C\}$ is consistent.■

There are several rules for replacing concepts and constraints which make life easier:

$\neg\forall R{:}C \quad \leftrightarrow \quad \exists R{:}\neg C$

$\neg\exists R{:}C \quad \leftrightarrow \quad \forall R{:}\neg C$

$x \in A \sqcap B \quad \leftrightarrow \quad x \in A, x \in B$

$$\begin{array}{lcl} x \in \forall R{:}C & \leftrightarrow & Rx \subseteq C \\ x \in \exists R{:}C & \leftrightarrow & y \in xR,\ y \in C \\ x \in (P = Q) & \leftrightarrow & xP = xQ \\ x \in \neg(P{=}Q) & \leftrightarrow & xP \neq xQ \end{array}$$

In order to show undecidability of subsumption, we use the undecidability of the word problem in groups [Bo59, No55]. Such a problem looks as follows: Let $R_1,\ldots,R_n$ be the symbols of a group and let $P_1{=}Q_1,\ldots,P_m{=}Q_m$ be the generating relations of some group. Then the word problem is to test given strings P andQ, whether P = Q is a consequence of these relations and the axioms for a group. We will show that there is a subsumption problem that is equivalent to this word problem.

In order to avoid clumsy notation, we assume that only associativity is built-in and that the relations imply that the semigroup defined by the relations is a group. If we have given generating relations under the assumption that all axioms for a group are built-in, the following procedure gives relations for semigroups ensuring that the generated semigroup is a group: Add a new symbol $R_e$ standing for the unit, and add for every symbol $R_i$ a new symbol $R_{\bar{i}}$ for the inverse. Then add the relations: $R_e \circ R = R$ and $R \circ R_e = R$ for every symbol $R \in \{R_e, R_i, R_{\bar{i}},\ i = 1,\ldots,n\}$ and add the relations $R_{\bar{i}} \circ R_i = R_e$ and $R_i \circ R_{\bar{i}} = R_e$ for all symbols $R_i$. The defining relations for the group are translated as follows: We assume that the words occurring in the relations are composed of symbols or inverses of symbols. The unit is translated into $R_e$, symbols are translated identically and inverses of symbols are translated by $(R_i)^{-1} \to R_{\bar{i}}$. Now the new relations together with the translated generated relations defining the group provide a semigroup, which is isomorphic to the original group and has an equivalent word problem.

These considerations permit us to assume in the following that the symbols are $R_1,\ldots,R_n$ and that the relations are $P_1{=}Q_1,\ldots,P_m{=}Q_m$, where the group defining relations are among these relations. By $\mathcal{G}$ we denote the free semigroup (which is in fact a group) generated by the symbols $R_1,\ldots,R_n$ with relations $P_1{=}Q_1,\ldots,P_m{=}Q_m$. For convenience we denote elements of $\mathcal{G}$ by [P], where P is a string of symbols and [P] denotes the congruence class with respect to the defining relations. Note that for two words it is semidecidable, whether they are congruent by replacement of equals by equals. Furthermore [P] = [Q] holds in $\mathcal{G}$, iff P = Q can be derived as equational consequence from the defining relations [BS81,Gr79].

We need a further role symbol R that is different from all other roles. Furthermore we use $P_i, Q_i$, P and Q also for the list of roles.

We define several concepts that are needed for the subsumption problem encoding the word problem. For convenience we use $\sqcap$ as associative operator and write lists of roles as composition.

$$C_1 := (R \circ R_1 = R) \sqcap \ldots \sqcap (R \circ R_n = R)$$
$$C_2 := \forall R{:}(P_1 = Q_1) \sqcap \ldots \sqcap \forall R{:}(P_m = Q_m)$$

Now let $C := C_1 \sqcap C_2$ and let $D := \forall R{:}\ (P{=}Q)$.

The idea of the construction is to view the relation in the role value maps as relations that define a semigroup and then to make deductions by replacing equals by equals. The concept $C_2$ encodes the relations that are used to define a particular group, wheras $C_1$ is only technical; it encodes some fixed point properties that will enable deductions as in groups.

Now we can prove the main result as a sequence of lemmas.

**3.2 Lemma.** D subsumes C, iff the following constraint system is inconsistent: $\{y \in xR, yP \neq yQ, xR \circ R_1 = R, \ldots, xR \circ R_n = xR, xR \subseteq (P_1 = Q_1), \ldots, xR \subseteq (P_m = Q_m)\}$.

**Proof.** Lemma 2.1 yields that D subsumes C iff $\neg D \sqcap C$ is an inconsistent concept, and Lemma 3.1 yields that this is equivalent to the inconsistency of the constraint system $\{x \in \neg D \sqcap C\}$. Due to the rules above, we can transform this constraint system into

$\{x \in \exists R{:}\neg(P=Q), x \in C_1, x \in C_2\}$ and thus into

$\{y \in xR, yP \neq yQ, x \in C_1, x \in C_2\}$.

If we develop the concepts $C_1$ and $C_2$, then we obtain the following constraint system:

$\{y \in xR, yP \neq yQ, xR \circ R_1 = R, \ldots, xR \circ R_n = xR, xR \subseteq (P_1 = Q_1), \ldots, xR \subseteq (P_m = Q_m)\}$.

■

In the constraint system obtained from Lemma 3.2 we abbreviate xR by s:

$$\{y \in s, yP \neq yQ, sR_1 = s, \ldots, sR_n = s, s \subseteq (P_1 = Q_1), \ldots, s \subseteq (P_m = Q_m)\}.$$

We denote this constraint system by $\mathcal{C}_\mathcal{G}$.

**3.3 Lemma.** If $P' = Q'$ is an equational consequence of the relations defining $\mathcal{G}$, then $s \subseteq (P' = Q')$ can be added to $\mathcal{C}_\mathcal{G}$ without affecting consistency.

**Proof.** The proof is by induction on a single deduction step. It suffices to prove this for the replacement of equals by equals. Let $A_i$ denote lists of roles. Let $\mathcal{C}$ be a constraint system obtained after the addition of some consequences to $\mathcal{C}_\mathcal{G}$. Let $s \subseteq (A_1 \circ A_2 \circ A_3 = A_4)$ and $s \subseteq (A_2 = A_5)$ be in $\mathcal{C}$.

We have to show that we can add the constraint $s \subseteq (A_1 \circ A_5 \circ A_3 = A_4)$ to $\mathcal{C}$. Therefore let M be a model and $\alpha$ be an assignment that satisfies $\mathcal{C}$ before the addition. For every element $a \in \alpha s$, we have $a\alpha(A_1 \circ A_2 \circ A_3) = a\alpha(A_4)$ and $a\alpha(A_2) = a\alpha(A_5)$. The latter implies that for every subset $B \subseteq \alpha s$ we have $B\alpha(A_2) = B\alpha(A_5)$. If we choose B as $\alpha A_1$, we obtain $a\alpha(A_1 \circ A_2) = a\alpha(A_1 \circ A_5)$. Obviously we can apply $\alpha A_3$ to the right. Thus we have $a\alpha(A_1 \circ A_2 \circ A_3) = a\alpha(A_1 \circ A_5 \circ A_3)$. This holds for every element a in $\alpha s$, hence $\alpha s \subseteq \alpha(A_1 \circ A_5 \circ A_3 = A_4)$. ■

**3.4 Lemma.** Let P and Q be words over the symbols generating $\mathcal{G}$. Then

$[P] = [Q]$ iff $\mathcal{C}_\mathcal{G}$ is inconsistent.

**Proof.**

"$\Rightarrow$": Lemma 3.3 shows that we can add add $s \subseteq (P = Q)$. which contradicts the constraints $y \in s, yP \neq yQ$. Hence $\mathcal{C}_\mathcal{G}$ is inconsistent.

"$\Leftarrow$": A model can be constructed as follows: Let the denotation of s be the set of congruence classes of the semigroup $\alpha s := \mathcal{G}$. The application of a role to $\alpha s$ is defined as $[P]R_i := [P \circ R_i]$. Then the constraints $sR_1 = s, \ldots, sR_n = s$ are satisfied, since $\mathcal{G}$ is a group. The constraints $s \subseteq (P_1 = Q_1), \ldots, s \subseteq (P_m = Q_m)$ are also satisfied, since the equations $[P_i] = [Q_i]$, $i = 1, \ldots, n$ hold in $\mathcal{G}$. It remains to be shown that $yP \neq yQ$, $y \in s$ can be satisfied. The assigment of the unit in $\mathcal{G}$ to y gives an element that is in the set $\alpha(P \neq Q)$, since $[P] \neq [Q]$ in $\mathcal{G}$. That s is an abbreviation for xR does not matter: we can simply add some element a to the model and define $\alpha x := a$ and $\alpha R := \{(a, b) \mid b \in \mathcal{G}\}$. Hence all constraints can be satisfied. ■

**3.5 Theorem.** Subsumption in KL-ONE*, and hence in KL-ONE, is undecidable.

**Proof.** Obviously the subsumption problem in Lemma 3.2 can be formulated in KL-ONE*. Lemmas 3.2, 3.3, 3.4 together with the well-known result that the word problem in groups is undecidable [Bo59, No55] shows the theorem. ■

## 4. Conclusion

We have shown that a considerable small sublanguage of KL-ONE has an undecidable subsumption problem. The reason for this nasty result seems to be the expressive power of role value maps. They are rather intuitive at first glance, but provide the full power of Turing machines if used excessively.

It is remarkable that role value maps in the feature term language [Sm88, SA87], where subsumption is co-NP-complete, do not have such a dramatic effect. The only difference between $\text{KL-ONE}_{+C}$ and the feature term language is that roles in the latter always are functional. The other possibility to restrict the expressive power of $\text{KL-ONE}_{+C}$ by discarding role value maps has been considered in [SS88], where it is shown that subsumption becomes co-NP-complete in this case.

Our conclusion is that either role value maps have to be discarded or simple restrictions on the use of role value maps have to be found, such that subsumption remains decidable.

## References.


AN86 Aït-Kaci, H., Nasr, R., LOGIN: A Logic Programming Language with built-in inheritance. J. Logic Programming **3**, 185-215, (1986)

Bo59 Boone, W.,W., The word problem, Ann. Math. (2), 70, 207-265, (1959)

BL84 Brachman R.J., Levesque,H.J.,The tractability of subsumption in frame-based description languages, Proc. AAAI-84, Austin, TX, (1984), pp.34-37

BS85 Brachman,R.J., Schmolze, J.G., An overview of the KL-ONE knowledge representation system. Cognitive Science 9(2), 171-216,(1985,

BFL83 Brachman, R.J., Fikes, R.E., Levesque, H.J., Krypton: Integrating Terminology and Assertion, Proc. AAAI-83, Washington DC, 1983, pp. 31-35

BGL85 Brachman, R.J., Gilbert, V.P., Levesque, H.J., An essential hybrid reasoning system: Knowledge and symbol level accounts of KRYPTON, Proc. of IJCAI-85, pp. 532-539, (1985) BS81 Burris, S., Sankappanavar, H.P., A course in universal algebra, Springer-Verlag, (1981)

Gr79 Grätzer, G., Universal Algebra, Springer-Verlag, (1979)

KBR86 Kaczmarek, T.S., Bates, R., Robins G., Recent developments in NIKL, Proc. AAAI-86, pp. 978-985, (1986)

LB85 Levesque, H,J, Brachman, R.J., A fundamental tradoff in knowledge representation, In Readings in Knowledge Representation, Morgan Kaufmann, (1985)

LB87 Levesque, H,J, Brachman, R.J., Expressiveness and tractability in knowledge representation and reasoning, Comput Intell **3**, 78-93, (1987)

LN87 Luck, K.v., Nebel, B., Peltason C., Schmiedel, A., The anatomy of the BACK system. KIT-report 41, Technische Universität Berlin, F.R. Germany, (1987)

MB87 MacGregor, R.M., Bates R., The Loom knowledge representation language, Technical report ISI/RS-87-188, Information Sciences Institute, Univ. of Southern California, (1987)

Mg88 MacGregor, R.M., A deductive pattern matcher, Proc. AAAI-88, (1988)

Ne88 Nebel, B., Computational complexity of terminological reasoning in BACK, Artifical Intelligence **34**, 371-383, (1988)

No55 Novikov, P.S., On the algorithmic undecidability of the word problem in group theory, Trudy Mat. Inst. Steklov. 14, Izdat. Nauk SSSR, Moscow, (1955)

PS84 Patel-Schneider, P.F., Small can be beautiful in knowledge representation, Proc. IEEE workshop on principles of knowledge based systems, Denver, Colorado, pp. 11-16, (1984)

PS88 Patel-Schneider, P.F., Undecidability of subsumption in NIKL, (to appear in J. of A.I., 1988)

Sc88 Schild, K., Undecidability of subsumption in $\mathcal{U}$, Draft, Institut für Software und theoretische Informatik, Technische Universität Berlin, (1988)

SS88 Schmidt-Schauß, M., Smolka, G., A frame-based language with complements, (to appear as SEKI-report , Universität Kaiserslautern, 1988)

Sh86 Shieber, S., An introduction to unification-based approaches to grammar, CSLI Lecture Notes 4, Stanford University, 1986

SA87 Smolka, G., Aït-Kaci, H., Inheritance hierarchies: Semantics and Unification. MCC Report AI-057-87, MCC, Austin, Texas, (1987)

Sm88 Smolka, G., A feature logic with subsorts, LILOG report 33, IBM Deutschland, Stuttgart, West Germany, (1988)

Vi85 Vilain, M., The restricted Language architecture of a hybrid representation system, Proc. of IJCAI-85, pp. 547-551, (1985)