Logical and Computational Properties of the Description Logic \textsc{Mirtl}\(^*\)

P. Buongarzoni, C. Meghini, R. Salis, F. Sebastiani and U. Straccia
Istituto di Elaborazione dell'Informazione
Consiglio Nazionale delle Ricerche
Via S. Maria, 46 - 56126 Pisa (Italy)
E-mail: \{\text{lastname}\}@iei.pi.cnr.it

1 Introduction

In recent years a number of positive (i.e. tractability and decidability) results have been found concerning the computational complexity of Description Logics (DLs) \cite{Buchheit et al.,1993; Donini et al.,1991; 1992a; 1992b: Schmidt-Schauß and Smolka,1991; Sebastiani and Straccia,1991}. Unfortunately, also negative results have appeared, e.g. showing that some DLs (e.g. NIKL \cite{Patel-Schneider,1989} and KL-ONE \cite{Schmidt-Schauß,1989}) are undecidable. This work contributes to this latter literature by showing a negative result for another DL (called \textsc{Mirtl}) which had not been previously studied from the standpoint of computational complexity, and which is not directly related to the ones already shown undecidable; in more standard DL terminology, \textsc{Mirtl} is the \textit{ALCQ}\textit{I} logic plus the \textit{T} operator for inverse roles and the \textit{O}\textit{I} operator for singleton concepts\(^1\).

We have recently been investigating the use of \textsc{Mirtl} for modelling multimedia information retrieval (see \cite{Meghini et al.,1993; Sebastiani,1994}). Given the requirements imposed by this application domain, and given the well-known negative results on the computational complexity of more powerful DLs, we had deemed \textsc{Mirtl} the best compromise between expressivity and tractability for our purposes. Somehow encouraged by the fact that the standard reasoning problems in the related logics \textit{ALCQ}\textit{IR} \cite{Buchheit et al.,1993}, \textit{ALCO} \cite{Schäfer,1994} and \textit{ALCN}\textit{T} (also known as \textit{PL}\textit{I}) \cite{Donini et al.,1991} are all decidable, and considering that the well-known algorithms based on constraint propagation for reasoning on them tend to be easily customizable to a chosen set of operators, we had thought that developing a sound and complete algorithm for \textsc{Mirtl} could be reasonably straightforward.

Unfortunately, while trying to develop such an algorithm, we discovered that \textsc{Mirtl} does not have the \textit{finite model property}; i.e. there are satisfiable \textsc{Mirtl} concepts (and assertions, and KBs) which are satisfiable only in interpretations of infinite cardinality. Although this result is not one of full-blown undecidability\(^2\), it however casts a shadow on the possibility of making practical use of such a logic. In fact, since the constraint propagation algorithms by now standard in the field of DLs prove the satisfiability of a concept by building a finite model of the concept, these algorithms are not applicable to \textsc{Mirtl} unless one renounces to guaranteed termination, or unless one builds some non-trivial loop-detecting control structure into them.

Once we abandon the idea of checking the satisfiability of a concept by building a finite model for it, the problem arises whether an alternative method exists or not. We have thus tried to find an upper bound to the complement of the satisfiability problem, i.e. to find a threshold under which possible inconsistencies should necessarily show up and above which no new inconsistency could emerge any more. This work reports on some negative results we have obtained in this direction, and which might constitute a prelude to a true undecidability result\(^3\).

2 \textsc{Mirtl} admits infinitary concepts

The language of \textsc{Mirtl} includes primitive concepts (\textit{A}), negation of primitive concepts (\textit{\neg A}), concept conjunction (\textit{C} \textit{\cap D}), universal quantification (\textit{\forall R.C}), qualified existential quantification (\textit{\exists R.C}), number restrictions (\textit{\geq n R} and \textit{\leq n R}), inversion of roles (\textit{R}\textit{\textasciicircum 1}) and singleton concepts (\textit{\{a\}}). We will also use the notation (\textit{\equiv n R}) in place of (\textit{\geq n R})\texttt{\cap}(\textit{\leq n R}), and the notation \textit{f(R,C)} in place of (\textit{\equiv 1 R})\texttt{\cap}(\textit{\forall R.C}). Finally, \textsc{Mirtl} allows assertions \textit{C[a]} and \textit{R[a,b]}, where \textit{C} is a concept, \textit{R} a role, and \textit{a} an individual.

\(^{*}\)This work has been carried out in the context of the project FERMI 8134 - "Formalization and Experimentation in the Retrieval of Multimedia Information", funded by the European Community under the ESPRIT Basic Research scheme.

\(^1\)This latter operator is a restricted form of the \textit{O}\textit{I} operator (also known as \textit{one-off}); the extension of a singleton concept \{\textit{a}\} is just the singleton containing the individual denoted by \textit{a}.

\(^2\)There are in fact decidable logics that do not have the finite model property; see e.g. \cite{Hughes and Cresswell,1984, page 154} or \cite{Yardi and Wolper,1986}.

\(^3\)Until recently we actually thought we had an undecidability proof for \textsc{Mirtl}, based on a reduction of the halting problem for a model of computation equivalent to Turing Machines; the proof was then shown to contain a mistake by Diego Calvanese, a mistake that we have not been able to fix yet.
Consider now the following MIRTL concept:

\[
\{ z \} \cap (\leq 0 \, S^{-1}) \cap \\
f(S), f(f(G^{-1}), \{ z \}) \cap \\
\forall G, f(S), ((\leq 1 \, S^{-1}) \cap f(G^{-1}), \{ z \})
\]  

(1)

The effect of concept (1) is to state some properties of the natural number zero (\( z \)); for instance, it states that \( z \) has no predecessor (\( S^{-1} \)) and a unique successor (\( S \)), and that all the numbers greater than \( z \) have a unique successor, which in turn is greater than \( z \) and has a unique predecessor.

**Theorem 2.1** Concept (1) admits only infinite models.

Concept (1) is satisfiable, as it admits the following infinite model:

\[
D = \{0, 1, 2, 3, \ldots \} \\
\{ z \} = \{0\} \\
S^{-1} = \{(0, 1), (2, 3), \ldots \} \\
G^{-1} = \{(0, 1), (0, 2), (0, 3), \ldots \}
\]

It can also be shown that if an interpretation \( I \) is a model of concept (1), then \( I \) must be defined on an infinite domain \( \{ z, d_1, d_2, d_3, \ldots \} \) such that the extension of \( G \) contains \( \{ (z, d_1), (z, d_2), \ldots \} \) and the extension of \( S \) contains \( \{ (z, d_1), (d_1, d_2), \ldots \} \).

This shows the existence of MIRTL concepts which are satisfied only in interpretations with infinite domain; we will call them **infinite concepts**. Similar concepts had already been shown to exist for other DLs, albeit substantially more expressive [Schilm, 1991; Calvanese et al., 1994].

The presence of infinite concepts has the practical consequence that the standard methods based on constraint propagation (e.g. the methods discussed in Schmidt-Schauf and Smolka, 1991; Buchheit et al., 1993) do not work properly; since these methods attempt to build a model of the concept they want to prove consistent, once applied to infinite concepts they loop forever, unless they are endowed with a control structure able to detect the construction of an infinitely self-replicating structure (which every MIRTL infinite concept, from what our experiments have shown, seems to contain).

The existence of MIRTL infinite concepts has a computational counterpart in what might be called **self-reactivating constraints**: as a consequence of the interaction among MIRTL propagation rules (in particular, the rules for universal quantification, role inversion and singleton), some constraints may be such that the rules that process them generate the same activating conditions that obtained before the application of the rule (thus inducing an infinite loop).

---

After having noticed this, in our search for a decidability result for MIRTL, we had hoped to find, nonetheless, an upper bound on the number of steps necessary to find a clash deriving from an inconsistent MIRTL concept. If we had found such an upper bound, say \( k \), decidability would have been established, as a MIRTL concept could be declared satisfiable after the \( k \)-th step had failed to produce a clash. Unfortunately, this strategy proved ineffective: for every \( f(n) \) (where \( n \) is the size of the concept) that we had, in repeated attempts, conjectured to be such an upper bound, we were able to discovered an unsatisfiable MIRTL concept which generated a clash well after \( f(n) \) steps. As we will see in the next section, this has happened for functions \( f \) of increasing order of magnitude.

### 3 Looking for an upper bound

Once we abandon the idea of checking the satisfiability of a concept by building a model for it, the problem arises whether an alternative method exists or not. To this end, we have studied the computational behaviour of various categories of infinite concepts. For instance, if we graphically represent a constraint system as a directed graph, in which nodes represent objects (individual constants or variables) and the set of concepts constraining them, and edges represent binary assertions between them (the edge is oriented from the first element of the assertion to the second), concept (1) generates a graph that contains an infinite subgraph like the one shown in Figure 1.

![Figure 1: (Partial) graphical representation of the constraints generated by Concept (1).](image-url)
repeat ad infinitum portions of the kernel subgraph. This shows that the constraint propagation process, after a finite number of de-activations of the same ψ-constraint, completes the construction of the kernel subgraph and starts building an infinite number of replicas of portions of it.

After observing this, we have tried to individuate the step of the constraint propagation process at which the building of the kernel subgraph is completed and the building of the replicas start. In fact, once this step were individuated, the problem would be solved; in fact, it would suffice to check if the kernel subgraph (plus a small number of replicas) contained an inconsistency, as, if this were not the case, also the replicas would not contain any inconsistency, and the concept could be deemed satisfiable without any further rule application.

In other words, what we have been looking for is an upper bound to the complement of the satisfiability problem (co-CS), i.e. a threshold under which possible inconsistencies should necessarily emerge and above which no new inconsistency could possibly emerge. We have tried to find such an upper bound as an upper bound on the number of de-activations of the same constraint, expressed in terms of the size $d$ of the initial concept. In fact, as we have already argued, the cause of termination of the computation is the re-activation of a constraint. Hence, we have made the hypothesis that, by limiting this number, one could limit the generation ad infinitum of replicas of portions of the kernel subgraph. We then started an empirical study in order to identify the maximum number of activations of the same constraint beyond which the procedure diverges and builds an infinite model.

### 3.1 Concepts denoting natural numbers (or: the $d^2$ limit)

In Section 2 we have seen that the procedure of constraint propagation diverges when applied to concept (1). By examining the constraint system generated after $h < d$ de-activations of the self-reactivating constraint

\[(\forall G.(f(S).((= 1 S^{-1}) \impliedby f(G^{-1}.\{z\}))))[z]\]

one can verify that no clash occurs in it. Within $d$ activations, inconsistencies due to arbitrarily nested concepts like the following, are discovered:

\[
(\forall R.(\exists P.(\exists R^{-1}.\{i\})) \impliedby
(\forall P^{-1}.(\forall P^{-1}.(\forall P^{-1}.(\forall P^{-1}.(\exists R) \impliedby \{i\})))) \impliedby (2)
\]

The peculiarity of this case comes from the depth of the syntactic tree of concept 2; the sub-concept $C'$ is to be found at depth level 7. In order for $C'$ to constrain an individual, it is necessary that the longest branch of the syntactic tree is completely explored. At this point, $C'$ constrains the individual constant $i$, and from the development of $C'[i]$ a clash may arise (e.g. if $C' = (\leq 0 R)$), or the infinite generation of variables may be blocked ($C' = (\leq n R)$), or still other things may happen.

### 3.2 Concepts denoting the multiples of a natural number (or: the $d^2$ limit)

The $d^2$ limit has resisted numerous empirical tests, until we have been able to find a MIRL concept denoting the set of the multiples of a natural number. For instance, the set of multiples of 2 may be represented by the following MIRL concept:

\[
(0) C_3 = (\{z\} \impliedby \\
(1) (\leq 0 E^{-1}) \impliedby \\
(2) (\leq 0 O^{-1}) \impliedby \\
(3) (\leq 0 E) \impliedby \\
(4) (\equiv 1 O) \impliedby \\
(5) (\exists M.\{t\}) \impliedby \\
(6) (\exists G.(\exists O^{-1}.\{z\})) \impliedby \\
(7)(7') (\forall G.((\forall O^{-1}.(\forall O.(\equiv 1 O^{-1}) \impliedby (\leq 0 O)) \impliedby (\leq 0 E^{-1}) \impliedby (\equiv 1 E)) \impliedby (\forall E.(\exists G^{-1}.\{z\})) \impliedby (\exists M.\{t\}))))) \impliedby \\
(7') (\forall E^{-1}.(\forall E.((\equiv 1 E^{-1}) \impliedby (\leq 0 E)) \impliedby (\leq 0 O^{-1}) \impliedby (\equiv 1 O)) \impliedby (\forall O.(\exists G^{-1}.\{z\}))))))
\]

If we interpret the roles $O, E, M, G$ as the “odd successor”, “even successor”, “multiples-of” and “bound above by”, one may see that $C_3$ denotes the set of even numbers; in fact, starting from the individual constant $z$ and generating variables $y_1, y_2, \ldots$, only variables with an even subscript are put in relation with the individual constant $t$. In fact, the first sub-concepts state that the natural number 0 (sub-concept 0) is neither the even successor of a number (1), nor its odd successor (2); also, 0 does not have any even successor (3), has exactly an odd successor (4), is a multiple of 2 (6). Sub-concept (7) (i.e. the one which generates the self-reactivating constraint) states that the odd successors of their predecessor have an even successor which is both a multiple of two and a positive number, while the even successors of their predecessors have an odd successor which is positive. In this way, a set of constraints is generated which is partially represented in Figure 2.

Each relation of type $M$ is generated after two activations of the constraint $(\forall G \ldots)$. In a similar way, it would be possible to describe the multiples of 3, 4, etc.  

Note that in the self-reactivating constraint (7) the disjunction of two concepts is simulated by means of two occurrences of the $\forall$ operator; in fact, of the two sub-concepts:

\[
(7') (\forall O^{-1}.(\forall O.(\ldots)))
\]

\[
(7') (\forall E^{-1}.(\forall E.(\ldots)))
\]
only one is activated when they are applied to the same variable. The activation depends on the subscript of the variable: variables with even subscript $k$ represent the even successor of the $k - 1$ variable; variables with odd subscript $k$ represent the odd successor of the same variable.

In order to describe the multiples of the numbers greater than 2 it is sufficient to modify concept $C_3$ by changing roles $E$ and $O$ into $S_0$ and $S_1$, and adding as many roles $S_j$ and as many subconcepts $(\forall S^{-1}_j \cdot (\forall S_j \cdot (\ldots)))$ as the number whose successors we want to represent ($S_j$ stands for “successor modulo $j$”).

This concept, which is in itself satisfiable, can be used in order to write unsatisfiable concepts whose first clash occurs after $d$ activations of a constraint. In fact, if the individual constant $t$ were constrained by concept $(\leq k \: M^{-1})$, for some $k$, the clash would be generated only after $nk - 1$ de-activations of the self-reactivating constraint, where $n$ is the number of subconcepts of type $(\forall S^{-1}_j \cdot (\forall S_j \cdot (\ldots)))$ of the self-reactivating constraint.

As the numbers $n$ and $k$ have the same order of magnitude of the dimension $d$ of the original concept, the first clash will appear in the constraint system before $O(d^2)$ steps but after $O(d)$ steps.

### 3.3 Concepts denoting the first $n$ powers of a number (or: the $d^d$ limit)

If we substitute, within concept 1 the self-reactivating subconcept:

\[(\forall G, f(S), \left(= 1 \: S^{-1}\right) \cap f(G^{-1}), \{z\})\]

by

\[(\forall G, f(S), \left(= 1 \: S^{-1}\right) \cap f(G^{-1}), \{z\}) \cap
(\exists R_1, ((\leq n \: R_1^{-1}) \cap
(\exists R_2, ((\leq n \: R_2^{-1}) \cap
(\exists R_3, ((\leq n \: R_3^{-1}) \cap
\{\{f\}\}))))))\]

we obtain an unsatisfiable concept, whose first clash appears within the constraint system after $d^2$ activations of the self-reactivating constraint. In fact, during the application of the propagation rules, the subconcept $(\exists R_1, \ldots)$ generates within the graph of Figure 3 a path of length 3 ending with the individual constant $t$. When a path for every variable $y_i$ is generated, a tree is created with root $t$ and depth 3. As the individual constant $t$ is constrained by concept $(\leq n \: R_3^{-1})$, when variable $y_{n+1}$ is generated (at the $n^2 + 1$-th deactivation of the $(\forall G \ldots)$ constraint), the first clash appears within the constraint system. Adding a role to path $R_1, R_2, \ldots$ corresponds to incrementing by one the depth of the tree, and consequentially incrementing by one the value $h$ of the exponent; instead, modifying the value $n$ of the number restrictions corresponds to modify the width of the tree, and consequentially modifying the base $n$. Obviously, both $h$ and $n$ are strictly smaller than $d$, but still remain of the same order of magnitude; hence, the number of de-activations of a constraint remains exponential in the dimension $d$ of the original concept $C$.

It is also possible to combine these two latter sources of complexity (multiples of a number ($d^d$) and powers of a number ($d^2$)), by generating the role $R_1$ only for the multiples of a number; this brings the maximum number of activations to $d \cdot d^d$.

### 4 Conclusions

In this work we have shown that MIRIL does not enjoy the finite model property, and have discussed some consequences of this fact. In particular, we have argued that, even if MIRIL should be proven decidable (which is still, to the best of our knowledge, an open problem), reasoning on it by means of constraint propagation al-

---

**Figure 2**: Graphical representation of the set of constraints denoting the multiples of 2.

**Figure 3**: Graphical representation of the constraints denoting the first power of 2.
algorithms is probably going to be computationally onerous, as MIRTL concepts may be written which generate clashes only after a computationally unfeasible number of steps.

Acknowledgements

We are grateful to Diego Calvanese for his detailed comments to an earlier draft (see Footnote 3), and for giving us interesting pointers to the literature.

References


