Standard and Non-Standard Inferences in the Description Logic $\mathcal{FL}_0$ Using Tree Automata (Abstract)[*]

Franz Baader, Oliver Fernández Gil, and Maximilian Pensel

Chair for Automata Theory, Theoretical Computer Science, TU Dresden, Germany
`firstname.lastname@tu-dresden.de`

In the early days of Description Logic (DL) research, the inexpressive DL $\mathcal{FL}_0$, which has only conjunction, value restriction and the top concept as concept constructors, was considered to be the smallest possible DL. It came as a surprise when it was shown that subsumption reasoning w.r.t. acyclic $\mathcal{FL}_0$ terminologies (TBoxes) is coNP-hard. The complexity increases when more expressive forms of TBoxes are used: for cyclic TBoxes to PSpace and for general TBoxes consisting of general concept inclusions (GCIs) even to ExpTime. Thus, w.r.t. general TBoxes, subsumption reasoning in $\mathcal{FL}_0$ is as hard as in $\mathcal{ALC}$, its closure under negation. In addition to standard inferences such as the subsumption and the instance problem, also various non-standard inferences, such as unification or computing the least common subsumer, were investigated for various DLs. For $\mathcal{FL}_0$, there are, however, very few results that consider general TBoxes. In the ExpTime-completeness result for subsumption in $\mathcal{FL}_0$ w.r.t. general TBoxes, the ExpTime upper bound is actually inherited from the more expressive DL $\mathcal{ALC}$. Dedicated subsumption algorithms and nonstandard inferences have been considered mostly for the case without TBoxes or w.r.t. a restricted form of (acyclic and cyclic) TBoxes in $\mathcal{FL}_0$ and its extension by number restrictions.

Here, we use automata working on infinite trees to solve both standard and non-standard inferences in $\mathcal{FL}_0$ w.r.t. general TBoxes.[1] On the one hand, we give an alternative proof of the ExpTime upper bound for subsumption in $\mathcal{FL}_0$ w.r.t. general TBoxes based on the use of looping tree automata. On the other hand, we employ parity tree automata to tackle the non-standard inference problems of computing the least common subsumer and the difference of $\mathcal{FL}_0$ concepts w.r.t. general TBoxes. To show our results, we basically proceed as follows:

1. We introduce the so-called least functional model of an $\mathcal{FL}_0$ concept w.r.t. an $\mathcal{FL}_0$ TBox, and prove that subsumption corresponds to inclusion of least functional models. Basically, a functional model is an infinite tree where every node has exactly one successor for every role. Such trees can be ordered by inclusion, i.e., by pairwise comparing the sets of concept names to which the nodes reached by the same role chain belong.

---

[1] Note that instead of automata one could use monadic second-order logic on trees, but the complexity upper bounds obtained this way would be worse than the ones we get using automata.

2. Then we show that such least functional models can be represented using so-called looping tree automata (LTAs): given an $\mathcal{FL}_0$ concept C and a general TBox $\mathcal{T}$, we can construct an LTA $\widehat{\mathcal{A}}_{C,\mathcal{T}}$ whose size is at most exponential in the size of C and $\mathcal{T}$ such that $\widehat{\mathcal{A}}_{C,\mathcal{T}}$ accepts exactly the least functional model of C w.r.t. $\mathcal{T}$. Using the product construction for LTAs, inclusion of least functional models can then be reduced to the emptiness problem for LTAs. Since the emptiness problem for LTAs is in P and the automata have at most exponential size, this yields the desired ExpTime upper bound for subsumption in $\mathcal{FL}_0$ w.r.t. general TBoxes. Note that, in contrast to the case of acyclic or cyclic TBoxes, where automata on finite or infinite *words* can be employed to decide subsumption, GCIs require the use of automata working on *trees*.

3. To deal also with non-standard inferences such as computing the least common subsumer (lcs) or the difference of two $\mathcal{FL}_0$ concepts w.r.t. a general $\mathcal{FL}_0$ TBox, the automata constructions need to be extended considerably. Instead of constructing an automaton that represents the least functional model of a fixed $\mathcal{FL}_0$ concept C, we now need to build an automaton accepting all trees representing least functional models of some $\mathcal{FL}_0$ concept w.r.t. $\mathcal{T}$. For this task, simple LTAs (which do not have an acceptance condition) are not sufficient; instead we use parity tree automata (PTAs), which use a set of so-called priorities in their acceptance condition. The construction of a PTA $\widehat{\mathcal{A}}_{\mathcal{T}}$ that accepts exactly the trees that are least functional models of some $\mathcal{FL}_0$ concept C w.r.t. a given general TBox $\mathcal{T}$ is quite involved. It makes use of the closure of PTAs under intersection, complement, and projection. The PTA $\widehat{\mathcal{A}}_{\mathcal{T}}$ constructed this way has double-exponentially many states and exponentially many priorities in the size of $\mathcal{T}$.

4. Given $\mathcal{FL}_0$ concepts C, D, their least common subsumer (lcs) w.r.t. a general TBox $\mathcal{T}$ is the the most specific concept that subsumes C and D w.r.t. $\mathcal{T}$. In general, the lcs need not exist if the TBox contains cyclic dependencies between concepts. We can show that it exists iff the intersection of the least functional models of C and D w.r.t. $\mathcal{T}$ is a least functional model of some $\mathcal{FL}_0$ concept w.r.t. $\mathcal{T}$. To decide whether the lcs exists, we can thus proceed as follows. We construct the automata $\widehat{\mathcal{A}}_{C,\mathcal{T}}$ and $\widehat{\mathcal{A}}_{D,\mathcal{T}}$, and use the product construction to obtain an automaton for the intersection of the least functional models of C and D. Using an additional product construction and the emptiness test for PTAs, one can then decide whether this intersection is accepted by $\widehat{\mathcal{A}}_{\mathcal{T}}$. The complexity of this decision procedure is in 2NExpTime $\cap$ co-2NExpTime.

5. The difference of $\mathcal{FL}_0$ concepts can be treated similarly, by using difference rather than intersection of least functional models.

Technical details and references to relevant work can be found in the full paper:

Baader, F., Fernández Gil, O., Pensel, M.: Standard and non-standard inferences in the description logic $\mathcal{FL}_0$ using tree automata. In: *GCAI 2018, 4th Global Conference on Artificial Intelligence*. EPiC Series in Computing 55, pages 1–14, EasyChair (2018).