Homework assignment 4 – Symbolic Systems I – UvA, June 2020

Ard Snijders

Assignment 1: modeling a knowledge base in ALC

TBOX:
***Person* is a category.**
$Person \in N_c$
Country* is a category, and *Republic* and *ConstitutionalMonarchy* are subcategories of *Country
$Country \in N_c$
$Republic \sqsubseteq Country$
$ConstitutionalMonarchy \sqsubseteq Country$
***City* is a category.**
$City \in N_c$
The categories of *persons*, *countries* and *cities* are disjoint.
$Person \sqcap Country \sqsubseteq \perp$
$City \sqcap Country \sqsubseteq \perp$
$City \sqcap Person \sqsubseteq \perp$
Republic and ConstitutionalMonarchy are disjoint
$Republic \sqcap ConstitutionalMonarchy \sqsubseteq \perp$
Every *country* is related by the relation *hasCapital* to some object in category *City.*
$Country \sqsubseteq \exists hasCapital.City$
Every country is related by the relation hasCapital to only objects in category City.
$Country \sqsubseteq \forall hasCapital.City$
Every country and every city is related by the role *hasHead* to some person.
$Country \sqsubseteq \exists hasHead.Person$
$City \sqsubseteq \exists hasHead.Person$
The categories *President*, *Monarch* and *Mayor* are subcategories of the category *Person.*
$President \sqsubseteq Person$
$Monarch \sqsubseteq Person$
$Mayor \sqsubseteq Person$
Everything in the category *ConstitutionalMonarchy* is related by the role *hasHead* to only things in *Monarch.*
$ConstitutionalMonarchy \sqsubseteq \forall hasHead.Monarch$
Everything in the category *City* is related by the role *hasHead* to only things in *Mayor.*
$City \sqsubseteq \forall hasHead.Mayor.$

ABOX:
***'NL'* is a country, and is a constitutional monarchy.**
NL : Country, NL : ConstitutionalMonarchy
'NL'* has as head *'WillemAlexander'.
(NL, WillemAlexander) : HasHead
'NL'* has as capital *'Amsterdam'.
(NL, Amsterdam) : HasCapital
'Amsterdam'* has as head *'Femke'.
(Amsterdam, Femke) : HasHead

See next page for remaining answer for assignment 1

Countries that are a republic cannot have a mayor as head.
This does not follow from the knowledge base; A country that is a republic can be related to some person by the role hasHead, and there is no GCI that restricts a person in Mayor to not also be related to a country via the role hasHead, thereby a republic can have a mayor as head; We'd have to add an extra piece of knowledge which states that for a country that is a republic, no objects related by the role hasHead are in the set Mayor.
ALC expression: $Countries \sqcap Republic \sqsubseteq \forall hasHead.\neg Mayor$.

'Femke' is a mayor.
ALC expression: $Femke : Mayor$. This follows from the knowledge base; (1) we know that 'Amsterdam' has as head 'Femke', (2) we know that 'NL' is a country, and is related to 'Amsterdam' by relation HasCapital, (3) we know that every country is related by the relation hasCapital to only objects in category City, (4) therefore it must follow from the knowledge base that 'Amsterdam' belongs to the category City, and we know that (5) everything in the category City is related by the role HasHead to only things in Mayor and (6) thus we can conclude that 'Femke' is a mayor.

'NL' is not a republic.
ALC expression: $NL : \neg republic$. This already follows from the knowledge base, since it was previously asserted that NL is a constitutional monarchy, and the sets constitutional monarchy and republic are disjoint, thus it follows from the KB that 'NL' is not a republic.

'WillemAlexander' is not a mayor.
ALC expression: $WillemAlexander : \neg Mayor$. This follows from the knowledge base, since (1) we know that 'NL' has as head 'WillemAlexander', (2) we know that 'NL' is a constitutional monarchy, (3) we know that everything in the subcategory ConstitutionalMonarchy is related by the role hasHead only with things in Monarch (4) from this it follows that 'WillemAlexander' cannot be a mayor, since if 'WillemAlexander' was a mayor, it would violate the GCI that everything related to ConstitutionalMonarchy via the role hasHead should be in Monarch only (which rules out the possibility of say 'WillemAlexander' being both a mayor and a monarch).

Assignment 2: encoding 3SAT into ALC satisfiability

We can convert a given 3CNF formula ϕ with ALC, by first converting its constituent clauses into complex concepts, after which we can take the intersection $\sqcap$ of these complex concepts to represent the conjunction of clauses in CNF. In order to translate a CNF clause to a complex concept, we can assign to each literal x_i a concept name A_i, and we can furthermore map to each logical operator in the propositional logic formula (disjunction and negation) their ALC-equivalent operators; for a disjunction between two literals, we can take the union between their concepts $\sqcup$, and for the negation of a literal we can use the complement of its concept with the $\neg$ operator. Following this procedure, we can express any given clause as a complex ALC concept, from which we can then construct C_ϕ by taking the intersection $\sqcap$ of these complex concepts.

It should be noted that for propositional logic, the task of determining satisfiability just concerns the (non)-existence of some truth assignment that satisfies the propositional formula. In ALC, the interpretation functions attempts to map every concept to a subset of its domain. Assuming our ALC-conversion from ϕ to C_ϕ, as described above, is correct, it follows that if such an interpretation function were to produce only an empty set $\emptyset$ for a concept C_ϕ, then this can be understood as the interpretation function being unable to map every concept to a subset of the domain; resultantly, we effectively only have a partial truth assignment, which is not suitable for satisfying a formula, and the absence of a non-empty set for the interpretation of C_ϕ^I thus indicates that ϕ is not satisfiable. Conversely, it should follow that if there exists an interpretation I for which $C_\phi^I \neq \emptyset$, then this means we have found a complete truth assignment (since every concept was mapped to some subset of the domain), and therefore it must be true that ϕ is satisfiable.

Assignment 3: knowledge base with only exponential models

We can construct a knowledge base KB_i, one for each positive integer $n \in 1, 2, 3, ...$, such that for each n, the knowledge base KB_n it holds that there are only interpretations I that satisfy all statements in the knowledge base whose domain δ^I is of size at least 2^n, using the following **TBOX** and **ABOX**:

TBOX
$A_{n-1} \sqsubseteq \exists r.B_{n,a} \sqcap A_n$.
$A_{n-1} \sqsubseteq \exists r.B_{n,b} \sqcap A_n$.
$B_{n,a} \sqcup B_{n,b} \sqsubseteq \bot$.
$B_{n,a} \sqsubseteq \forall r.B_{n,a}$.
$B_{n,b} \sqsubseteq \forall r.B_{n,b}$.

ABOX
$x : A_0$ (Root node of tree)

The first two lines in the TBOX state that every item in A_{n-1} must be related by relation r to some object in the union of $B_{n,a}$ and A_n, and the union of $B_{n,b}$ and A_n. We then declare all sets $B_{n,a}$ and $B_{n,b}$ to be disjoint with the last line in the TBOX. This ensures that for a given value of n, there will always be a minimum of 2^n solutions. The last two lines state that every object in $B_{n,a}$ should be related by relation r to only objects in $r.B_{n,a}$.