Logics and categories for software engineering and artificial intelligence

Till Mossakowski, Lutz Schröder — University of Bremen
Summer Semester 2009 — Department of Computer Science

Exercise Sheet 3

Due: May 19, 2009

Exercise 3.1 (Conservative Extensions)

Consider your solution to Exercise 1.3 from the last exercise sheet, and consider the theory morphism $\sigma : (\Sigma_1, \Gamma_1) \to (\Sigma_2, \Gamma_2)$, where

$$\begin{aligned} \Sigma_1 &= \{\mathit{black\_exhaust}, \mathit{blue\_exhaust}, \mathit{low\_power}, \mathit{overheat}, \mathit{ping}, \\ &\quad \mathit{incorrect\_timing}, \mathit{clogged\_filter}, \mathit{low\_compression}, \mathit{carbon\_deposits}, \\ &\quad \mathit{clogged\_radiator}, \mathit{defective\_carburetor}, \mathit{worn\_rings}, \mathit{worn\_seals}\}, \\ \Sigma_2 &= \Sigma_1 \cup \{\mathit{replace\_auxiliary}, \mathit{repair\_engine}, \mathit{replace\_engine}\}, \end{aligned}$$

Γ_1 contains all the axioms corresponding to the symptoms (the overheating engine and the fact that the ignition timing is correct) as well as all the axioms describing diagnostic rules (i.e., the formalizations of facts (i) through (vi) in the informal description in Exercise 1.3). Γ_2 contains all axioms from Γ_1 plus the three rules corresponding to facts (vii) through (ix). The morphism σ is the inclusion mapping from Σ_1 into Σ_2 mapping each proposition to itself.

(a) Show that σ is a model-theoretically conservative theory morphism.

(b) Reformulate your HETS specification such that (Σ_2, Γ_2) is specified as an extension to (Σ_1, Γ_1) using the **then** keyword. Additionally, indicate that the extension is supposed to be conservative using **%cons**. Use HETS to prove that this is indeed the case (you will need the latest nightly build of HETS to do that[1]).

Exercise 3.2 (Description Logics)

Note: This exercise is not graded.
Familiarize yourself with the *pizza ontology*.
It can be found at `http://www.co-ode.org/ontologies/pizza/`.

Exercise 3.3 (Specification extensions)

Construct a specification extension such that the basis specification has models with 0, 1, 2 and 3 different expansions.

Exercise 3.4 (Conservative extensions)

[1] You can download the new HETS library from the lecture wiki (Resources/Software) and follow the installation instructions provided there.

Consider the following specifications.

(i)

logic PROPOSITIONAL
spec BLOCKSHAPES =
  props *cube tetrahedon*
  • $cube \lor tetrahedon$
  • $\neg(cube \land tetrahedon)$
then %%cons?
  prop *dodecahedron*
  • $cube \lor tetrahedon \lor dodecahedron$
  • $\neg(cube \land dodecahedron)$
  • $\neg(tetrahedon \land dodecahedron)$
  • $\neg cube \Rightarrow dodecahedron$
end

(ii)

logic PROPOSITIONAL
spec IMPLICATIONS =
  props a b c
  • $a \Rightarrow b$
  • $b \Rightarrow c$
then %%cons?
  prop d
  • $\neg(d \Rightarrow a)$
end

(a) Decide whether the extensions in (i) and (ii) are conservative. If they are not, provide models that cannot be expanded.

(b) In case of lacking conservativity, use the theorem from the lecture to construct from the model a sentence that can be proven in the extended theory, but not in the base theory.

Exercise 3.5 (Pizza ontology)

Formalize the following statements from the pizza ontology in Hets:[2]

- Pizza is food.
- Pizza base is food.
- Pizza topping is food.
- Pizzas, pizza bases, and pizza toppings are disjoint sets of things.
- A fish topping is a pizza topping.
- A meat topping is a pizza topping.
- Pizzas have pizza toppings.
- Pizzas have unique pizza bases.
- Thin and crispy pizza bases are pizza bases.
- A thin and crispy pizza is a pizza that only has a thin and crispy base.
- An interesting pizza is a pizza that has at least three toppings.
- A vegetarian pizza is a pizza which has neither a meat topping nor a fish topping.

Exercise 3.6 (Deductive ontology)

Download and read the document describing the deductive ontology introduced in the lecture.[3]

[2] The input format follows the Manchester syntax. To get an idea of what such a formalization might look like, you can review the formalization of the family ontology from the lecture at `http://www.informatik.uni-freiburg.de/~ki/teaching/ws0809/lccai/family.het`.

[3] `http://www.cs.miami.edu/~tptp/cgi-bin/SeeTPTP?Category=Documents&File=SZSOntology`

Formalize the part of the deductive ontology describing the concepts

- `Satisfiable`,
- `Theorem`,
- `WeakerTheorem`,
- `Equivalent`,
- `TautologousConclusion`,
- `EquivalentTheorem`,
- `Tautology`,
- `ContradictoryAxioms`,
- `SatisfiableConclusionContradictoryAxioms`,
- `TautologousConclusionContradictoryAxioms`, and
- `NoConsequence`,

i.e., the left half of the graphic depicting the deductive ontology, using Manchester syntax. Follow these steps:

(a) Introduce basis concepts describing the status of the axioms (valid, satisfiable, unsatisfiable), the status of the conjecture (valid, satisfiable, unsatisfiable), and the possible entailment relations between the axioms and the conjecture (all models of the axioms are models of the conjecture, some models of the axioms are models of the conjecture, etc).

(b) For these basis concepts, formalize all subsumption, equivalence and disjointness relations that you are aware of.

(c) Define the eleven concepts listed above as intersections of (complements of) basis concepts. Follow the definitions of the concepts given in the Section *Deductive Statuses* of the document describing the deductive ontology.

The exercise sheets may and should be worked on in groups of two (2) students. Please write both names on your solution.