National University of Singapore
School of Computing
CS3243 Introduction to AI

Tutorial 9: Inference in First-Order Logic

Issued: April 23, 2020 Due: Week 12, in the tutorial class

Important Instructions:

- *Your solutions for this tutorial must be TYPE-WRITTEN.*
- *Make TWO copies of your solutions: one for you and one to be submitted in class/online. Late submission will NOT be entertained.*
- *You may discuss the content of the questions with your classmates, but you should work out and write up ALL the solutions by yourself.*
- *This tutorial is UNGRADED, but do expect to see similar questions in the final exam.*

1. (Question 9.4 from AIMA 3rd edition) For the following pairs of atomic sentences, give the most general unifier if one exists:

   (a) $P(A, B, B)$, $P(x, y, z)$

   Solution: $\{x \leftarrow A, y \leftarrow B, z \leftarrow B\}$

   (b) $Q(y, G(A, B))$, $Q(G(x, x), y)$

   Solution: Failure. Note that x cannot bind to both A and B.

   (c) $\texttt{Older}(\texttt{Father}(y), y)$, $\texttt{Older}(\texttt{Father}(x), \texttt{John})$

   Solution: $\{x \leftarrow \texttt{John}, y \leftarrow \texttt{John}\}$ or $\{x \leftarrow \texttt{John}, y \leftarrow x\}$. Note that the latter is produced by the unification algorithm. Recall that the MGU is unique up to renaming and substitution of variables.

   (d) $\texttt{Knows}(\texttt{Father}(y), y)$, $\texttt{Knows}(x, x)$

Solution: Failure. Note that $\text{OCCUR-CHECK?}(y, \texttt{Father}(y))$ prevents unification of y with $\texttt{Father}(y)$.

2. An economist asks two knowledge engineers to write the sentence "Nothing is free." The first engineer wrote $\neg\exists x : \texttt{Free}(x)$ and transformed it into the following clause:

$$\neg\texttt{Free}(G1)$$

where $G1$ is a Skolem constant. The second engineer wrote $\forall x : \neg\texttt{Free}(x)$ and transformed it into the following clause:

$$\neg\texttt{Free}(x)$$

Why did they produce two different versions? Which version is correct?

Solution: The second version is correct and the first version is wrong. The first engineer made a mistake by skolemizing the sentence before moving $\neg$ inwards.

3. Two English sentences "Anyone who takes an AI course is smart" and "Any course that teaches an AI topic is an AI course" have been represented in first-order logic:

$$\forall x : (\exists y : \texttt{AI\_Course}(y) \land \texttt{Takes}(x, y)) \Rightarrow \texttt{Smart}(x)$$

$$\forall x : (\exists y : \texttt{AI\_Topic}(y) \land \texttt{Teaches}(x, y)) \Rightarrow \texttt{AI\_Course}(x)$$

It is also known that John takes CS3243 and CS3243 teaches Inference, which is an AI topic. Represent these facts as first-order logic sentences. Now convert all first-order logic sentences into conjunctive normal form and use resolution to prove that "John is smart."

Use the following table format to represent the steps you take. Here, sentences 1 and 2 are the ones you are resolving. $\theta =$ is the substitutions you are making to unify them (if any). The resolvent is (as its name suggests) the sentence that results from resolving the two sentences you picked. For brevity, you can use the number of the sentence instead of copying verbatim. For example, if my KB has two facts (1) and (4) as clauses I want to resolve, I can write (1) in sentence 1, (2) in sentence 2, and assign a new number to the resolvent (if I want to use it in the future).

Step #	Sentence 1	Sentence 2	Resolvent	$\theta =$

Solution: Convert 1st sentence to CNF:

$$\forall x : \neg(\exists y : \texttt{AI\_Course}(y) \wedge \texttt{Takes}(x,y)) \vee \texttt{Smart}(x)$$
$$\forall x : \forall y : \neg\texttt{AI\_Course}(y) \vee \neg\texttt{Takes}(x,y) \vee \texttt{Smart}(x)$$
$$\neg\texttt{AI\_Course}(y) \vee \neg\texttt{Takes}(x,y) \vee \texttt{Smart}(x) \tag{1}$$

Convert 2nd sentence to CNF:

$$\forall x : \neg(\exists y : \texttt{AI\_Topic}(y) \wedge \texttt{Teaches}(x,y)) \vee \texttt{AI\_Course}(x)$$
$$\forall x : \forall y : \neg\texttt{AI\_Topic}(y) \vee \neg\texttt{Teaches}(x,y) \vee \texttt{AI\_Course}(x)$$

Standardize variables apart from the 1st sentence:

$$\forall v : \forall u : \neg\texttt{AI\_Topic}(u) \vee \neg\texttt{Teaches}(v,u) \vee \texttt{AI\_Course}(v)$$
$$\neg\texttt{AI\_Topic}(u) \vee \neg\texttt{Teaches}(v,u) \vee \texttt{AI\_Course}(v) \tag{2}$$

Facts:
(3) : $\texttt{Takes}(\texttt{John}, \texttt{CS3243})$
(4) : $\texttt{Teaches}(\texttt{CS3243}, \texttt{Inf})$
(5) : $\texttt{AI\_Topic}(\texttt{Inf})$

Added negated goal for proof by contradiction:
(6) : $\neg\texttt{Smart}(\texttt{John})$

Resolution:

No.	Sentence 1	Sentence 2	Resolvent	$\theta =$
(7)	(1)	(6)	$\neg\texttt{AI\_Course}(y) \vee \neg\texttt{Takes}(\texttt{John}, y)$	$\{x \leftarrow \texttt{John}\}$
(8)	(3)	(7)	$\neg\texttt{AI\_Course}(\texttt{CS3243})$	$\{y \leftarrow \texttt{CS3243}\}$
(9)	(2)	(8)	$\neg\texttt{AI\_Topic}(u) \vee \neg\texttt{Teaches}(\texttt{CS3243}, u)$	$\{v \leftarrow \texttt{CS3243}\}$
(10)	(5)	(9)	$\neg\texttt{Teaches}(\texttt{CS3243}, \texttt{Inf})$	$\{u \leftarrow \texttt{Inf}\}$
(11)	(4)	(10)	[]	$\{\}$

4. PSA would like to implement its tax system on ships and cargo for its Brani port as part of a first-order logic system. You have been hired as a knowledge engineer to convert the following predicates into FOL representation. You may use any predicate that you create in previous parts for subsequent parts. You may also define new constants and predicates.

   Note that variables are in lowercase, whereas constant, predicate and function symbols start with uppercase. Given the functions:

   $\texttt{Arrival\_Time}(ship)$
   $\texttt{Departure\_Time}(ship)$

And the predicates:

$$\texttt{Unload\_From\_Ship}(cargo, ship, arrival\_time)$$
$$\texttt{Load\_Onto\_Ship}(cargo, ship, departure\_time)$$
$$\texttt{Weapons}(cargo)$$

(a) In the PSA system, can ship objects (n.b., not constant symbols) arrive at the Brani port multiple times? Justify your answer.

Solution: Yes, they can, but the same ship object will have to be referred to using different constant symbols in the interpretation. This is necessary as the arrival and departure times are specified as functions, which always return the same object.

(b) Write a FOL predicate of a simplified tax law for cargos entering and not departing Singapore. Note that ships unload and load at their arrival and departure times.

Solution: We'd like to state that any cargo that ends up staying in Singapore should be taxed. More formally (but still in words), for any piece of cargo c, if it gets unloaded at some point in time from some ship s_1 and does not get loaded onto any other ship s_2 at any other point of time, then c is taxable.

$$\forall c :((\exists s_1 : \texttt{Unload\_From\_Ship}(c, s_1, \texttt{Arrival\_Time}(s_1)))\wedge$$
$$(\neg\exists s_2 : \texttt{Load\_Onto\_Ship}(c, s_2, \texttt{Departure\_Time}(s_2)))) \Rightarrow \texttt{Taxable\_Cargo}(c)$$

(c) Aside from cargo, ships are also taxed. A ship is taxable upon entry to Singapore unless all the ship's cargo are weapons (slated for the armed forces).

Solution: In words - if a ship arrives and its cargo is not weapons, then the ship should be taxed.

$$\forall s : (\exists c : \texttt{Unload\_From\_Ship}(c, s, \texttt{Arrival\_Time}(s))\wedge\neg\texttt{Weapons}(c)) \Rightarrow \texttt{Taxable}(s)$$

(d) State your answer from Part (c) in Conjunctive Normal Form.

Solution:

$$\neg\texttt{Unload\_From\_Ship}(c, s, \texttt{Arrival\_Time}(s)) \vee \texttt{Weapons}(c) \vee \texttt{Taxable}(s)$$

(e) Using the following observations, use resolution by refutation to answer the query $\texttt{Taxable}(\texttt{Storm\_King})$:

- At its arrival time, torpedos were unloaded from the ship "Storm King"
- Torpedos are weapons.
- At its arrival time, laser parts were unloaded from the ship "Storm King"
- Laser parts are not weapons.

No.	Sentence 1	Sentence 2	Resolvent	$\theta =$
(1)	part (d)	$\neg\texttt{Tax}(\texttt{SK})$	$\neg\texttt{UFShip}(c, \texttt{SK}, \texttt{ATime}(\texttt{SK})) \vee \texttt{Weapons}(c)$	$\{s \leftarrow \texttt{SK}\}$
(2)	(1)	$\texttt{UFShip}(\texttt{LP}, \texttt{SK}, \texttt{ATime}(\texttt{SK}))$	$\texttt{Weapons}(\texttt{LP})$	$\{c \leftarrow \texttt{LP}\}$
(3)	(2)	$\neg\texttt{Weapons}(\texttt{LP})$	[]	$\{\}$

Solution: The known facts can be written as

$$\texttt{Unload\_From\_Ship}(\texttt{Torpedos}, \texttt{Storm\_King}, \texttt{Arrival\_Time}(\texttt{Storm\_King}))$$
$$\texttt{Weapons}(\texttt{Torpedos})$$
$$\texttt{Unload\_From\_Ship}(\texttt{Laser\_Parts}, \texttt{Storm\_King}, \texttt{Arrival\_Time}(\texttt{Storm\_King}))$$
$$\neg\texttt{Weapons}(\texttt{Laser\_Parts})$$

We wish to prove that $\texttt{Taxable}(\texttt{Storm\_King})$, so we add the sentence $\neg\texttt{Taxable}(\texttt{Storm\_King})$ to our Knowledge Base and resolve until we arrive at an empty clause. We abbreviate

i. `Storm_King` to `SK`
ii. `Laser_Parts` to `LP`
iii. `Taxable` to `Tax`
iv. `Arrival_Time` to `ATime`
v. `Unload_From_Ship` to `UFShip`

so the resolution table will fit the text margins.