London Metropolitan University

CS7051- Semantic Technologies

Coursework –1

Individual Report

Submitted By:

Divya Prackash Ravi | 22014173

Submitted To:

Dr. Vassil Vassilev

Date: March 19th, 2023

TABLE OF CONTENTS

1. Satisfiability

<u>Fact :</u>

The different Airports which are called {A1,A2,A3,A4,A5,A6} in the model

A1= Reykjavik, A2= Dublin, A3= Kiev, A4= Vienna, A5= Oslo, A6 = Rome

|| Reykjavik || = A1

Airport : { <A1>,<A2>,<A3>,<A4>,<A5>,<A6>}

Airport(x) where x is the variable

Substitute x with the logical Constant Reykjavik, we get Airport(Reykjavik) which states that the Reykjavik is an airport.

< Reykjavik > ∈ || Airport ||

Therefore the fact Airport is satisfied

<u>Universal heuristic :</u>

Every plane in journey has a passenger
$\forall x\ \forall y\ Planes(x) \cap Journey(y) \cap Plane\_in\_Journey(x,y) \rightarrow \exists z\ Passenger(z) \cap Person\_in\_Plane(x,z)$

Passenger : {PS1,PS2,PS3,PS4,PS5,PS6,PS7,PS9,PS10}
Planes : {P1,P2,P3,P4}
Journey : {J1, J2, J3, J4, J5}

Person_in_Plane: { <Ps1,P1> , <Ps2,P1> , <Ps3,P1> , <Ps4,P1> , <Ps5,P1> , <Ps6,P2> , <Ps7,P2> , ……}
Plane : {<P1>,<P2>,<P3>,<P4>}
Journey: {<J1>,<J2>,<J3>,<J4>,<J5>}
Passenger: {<PS1>,<PS2>,………………<PS10>}
Plane_in_Journey: {<P1,J1>,<P10,J3>,<P5,J2>…………..}

By substituting the respective values given above we can retrieve fact stating that the Plane x on a Journey y has at least one Passenger z is in the plane

Existential Heuristic :

Some journeys do not have direct route
$\exists$ x Journey (x) $\rightarrow$ $\exists$y$\exists$z Airports (y) $\cap$ Airports (z) $\cap$ ($\neg$ Direct_Route(y,z))

Journey : {J1,J2,J3,J4,J5 }

Airports : { A1, A2, A3, A4, A5,A6}

Journey : {<J1>,<J2>,<J3>,<J4>, <J5>}
Airports: {<A1>,<A2>,<A3>,<A4>,<A5>,<A6>}
Direct_Route : {<A1,A3>,<A4,A5>,………………}

In this existential heuristic we say that there exists one journey wherein there exist atleast one set of airport which has no direct route from starting airport to ending
In the Journey J5 it goes from A3 to A1 via A5 and A4 thus satisfying the rule.

2. Deducing Conclusions

Deducing conclusions in your theory using 3 inference rules
The Airlines AL3 does not operate in Route R6

1.	Route(R6)	Fact
2.	$\exists x$ Route(x) $\rightarrow$ Journey(x)	Heuristic
3.	Route (R6) $\rightarrow$ Journey(R6)	1, EI
4.	Journey(R6)	1,3, MP
5.	Airlines(AL3)	Fact
6.	$\exists x$ Airlines(x) $\rightarrow \exists y \neg$Journey(y)	Heuristic
7.	$\neg$Airlines(AL3)	4,5,6, MT

Therefore we come to the conclusion that the Journey taken on R6 is not operated by Airlines AL3.

3. Conversion to Horn-Clause

Facts :

Airport (A4)

Clausal Form : {Airport (A4)}

Plane (P1)

Clausal Form :{Aeroplane (P1)}

Airline (AL3)

Clausal Form : {Airline (AL3)}

Person_in_Plane (P1, PS2)

Clausal Form : { Person_in_Plane (P1, PS2)}

Heuristics :

i. Every plane in a journey has a passenger

∀x ∀y Planes(x) ∩ Journey(y) ∩ Plane_in_Jouney(x,y) → ∃z Passenger(z) ∩ Person_in_Plane(x,z)

Eliminate Implication :
∀x ¬∀y (Planes(x) ∩ Journey(y) ∩ Plane_in_Jouney(x,y)) U (∃z Passenger(z) ∩ Person_in_Plane(x,z))

Move negation in :
∀x ∃y ¬Planes(x) U ¬Journey(y) U ¬Plane_in_Jouney(x,y) U (∃z Passenger(z) ∩ Person_in_Plane(x,z))

Skolemise y and z :
∀x ¬Planes(x) U ¬Journey(F(x)) U ¬Plane_in_Jouney(x,F(x)) U (Passenger(G(x)) ∩ Person_in_Plane(x,G(x)))

Distribute disjunctions :
∀x (¬Planes(x) U ¬Journey(F(x)) U ¬Plane_in_Jouney(x,F(x)) U

Passenger(G(x)) ∩ (¬Planes(x) U ¬Journey(F(x)) U ¬Plane_in_Jouney(x,F(x)) U Person_in_Plane(x,G(x)))

Drop Quantifiers :
(¬Planes(x) U ¬Journey(F(x)) U ¬Plane_in_Jouney(x,F(x)) U Passenger(G(x)) ∩ (¬Planes(x) U ¬Journey(F(x)) U ¬Plane_in_Jouney(x,F(x)) U Person_in_Plane(x,G(x)))

Clausal Form :
{
{¬Planes(x) , ¬Journey(F(x)) , ¬Plane_in_Jouney(x, F(x)) , Passenger(G(x))},
{¬Planes(x) , ¬Journey(F(x)) , ¬Plane_in_Jouney(x, F(x)) , Passenger(G(x)}
}

ii. Some airline do not do specific journeys

∃x Airline(x) → ∃y ¬Journey(y)

Eliminate Implication :
¬∃x Airline(x) U ∃y ¬Journey(y)

Move negation in :
∀x ¬Airline(x) U ∃y ¬Journey(y)

Skolemise y :
∀x ¬ Airline(x) U ¬Journey (F(x))

Drop Quantifiers :
¬ Airline(x) U ¬Journey (F(x))

Clausal Form :
{¬Airline(x), ¬Journey (F(x))}

iii. Some journeys do not have a direct route
∃ x Journey (x) → ∃y∃z Airports (y) ∩ Airports (z) ∩ (¬ Direct_Route(y,z))

Eliminate Implication :
¬ ∃ x Journey (x) U (∃y∃z Airports (y) ∩ Airports (z) ∩ (¬ Direct_Route(y,z)))

Move negation in :
∀ x ¬ Journey(x) U (∃y∃z Airports (y) ∩ Airports (z) ∩ (¬ Direct_Route(y,z)))

Skolemise y and z :

$\forall$ x ¬ Journey(x) U (Airports (F(x)) ∩ Airports (z) ∩ (¬ Direct_Route(y,z)))

<u>Distribute disjunctions :</u>
$\forall$ x (¬ Journey(x) U Airports (F(x))) ∩ (¬ Journey(x) U Airports (z)) ∩ (¬ Journey(x) U (¬ Direct_Route(y,z)))

<u>Drop Quantifiers :</u>
(¬ Journey(x) U Airports (F(x))) ∩ (¬ Journey(x) U Airports (z)) ∩ (¬ Journey(x) U (¬ Direct_Route(y,z)))

<u>Clausal Form :</u>
{
{¬ Journey(x) , Airports (F(x))},
{¬ Journey(x) , Airports (z)},
{¬ Journey(x) ,¬ Direct_Route(y,z)}
}

iv. Some routes form a journey by itself
∃x Route(x) → ∃y Journey(y)

<u>Eliminate Implication :</u>
¬∃x Route(x) U ∃y Journey(y)

<u>Move negation in :</u>
$\forall$ x ¬Route(x) U ∃y Journey(y)

<u>Drop universal quantifier :</u>
¬Route(x) U ∃y Journey(y)

<u>Skolemise y :</u>
¬Route(x) U Journey(F(x))

<u>Clausal form :</u>
{¬Route(x) , Journey(F(x))}

4. Resolution

Does the Airline z operate on Route r

Some routes form a journey by itself

1 ¬Route(x) U Journey(y) premise

Some Airlines do not do specific journeys

2 ¬ Airline(z) U ¬Journey (y) premise

Some Airlines do not use specific routes

3 ¬Route(x) U ¬Airline(z) 1,2, Res