SESSION 11

Exercises
PDF-dokument · 1,8 MB

1. Consider the following information.
   - If the unicorn is mythical, then it is immortal, but if it is not mythical, then it is a mortal mammal.
   - If the unicorn is immortal or a mammal, then it is horned.
   - The unicorn is magical if it is horned.

   (a) Can you prove that the unicorn is mythical? How about magical? Horned?
   (b) Write the facts in propositional logic.
   (c) Write the rules in propositional logic (use only rules of the form $a \Rightarrow b$ or $a \Rightarrow b$ or $a \Rightarrow b$ or $a \Rightarrow b$)
   (d) Translate the formulas you wrote into CNF and apply resolution to conclude your results.

a) we cant know abt Mythical
all are magical
all are horned

b) $(mythical \Rightarrow immortal) \land \neg(mythical \Rightarrow immortal \land mammal)$ → $mythical \rightarrow Immortal$; $\neg mythical \rightarrow (\neg Immortal \land mammal)$

$(immortal \lor mammal) \Rightarrow horned$
$horned \rightarrow magical$

c) if we want to do fc, it must be in horneform

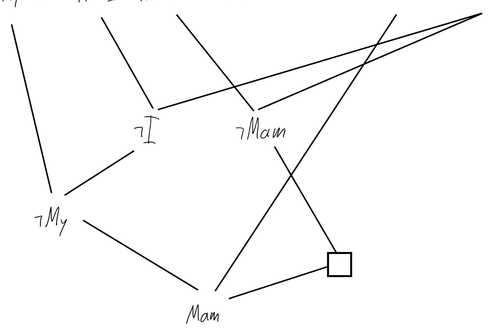
d) $Mythical \rightarrow Immortal \equiv$
$\neg Mythical \lor Immortal$

$(Immortal \lor mammal) \rightarrow horned \equiv$
$\neg(Immortal \lor mammal) \lor horned \equiv$
$(\neg Immortal \land \neg mammal) \lor horned \equiv$
$(\neg Immortal \lor horned) \land (\neg mammal \lor horned)$

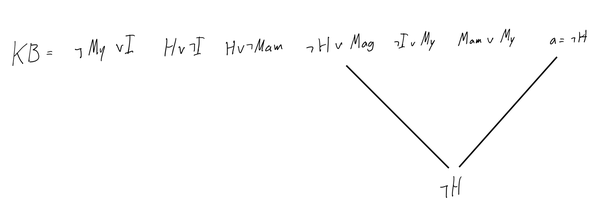
$\neg horned \lor magical$

$mythical \lor (\neg Immortal \land mammal)$
$(mythical \lor \neg Immortal) \land (mythical \lor mammal)$

d2) $KB = \neg My \lor I \quad H \lor I \quad H \lor \neg Mam \quad \neg H \lor Mag \quad \neg I \lor My \quad Mam \lor My \quad a = \neg H$



$KB = \neg My \lor I \quad H \lor I \quad H \lor \neg Mam \quad \neg H \lor Mag \quad \neg I \lor My \quad Mam \lor My \quad a = \neg H$

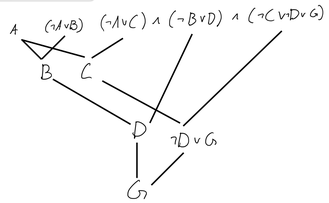


16. Consider the following sentence:
$$(Food \Rightarrow Party) \lor (Drinks \Rightarrow Party) \lor (Food \land Drinks) \Rightarrow Party$$

a) $((F \rightarrow P) \lor (D \rightarrow P)) \rightarrow ((F \land D) \rightarrow P)$

18. Consider the following conjunction of clauses,
$$(A \lor \neg A \lor B) \land (\neg A \lor C) \land (\neg B \lor D) \land (\neg C \lor \neg D \lor G)$$
Prove that the above sentence entails G using:
(a) resolution,
(b) forward chaining,
(c) backward chaining.

a) $KB = A \quad (\neg A \lor B) \quad (\neg A \lor C) \land (\neg B \lor D) \land (\neg C \lor D \lor G) \quad a = G \quad a_2 = \neg G$



b) A
$(\neg A \lor B) \equiv A \rightarrow B$
$(\neg A \lor C) \equiv A \rightarrow C$
$(\neg B \lor D) \equiv B \rightarrow D$
$(\neg C \lor \neg D \lor G) \equiv \neg(C \lor D) \lor G \equiv (C \lor D) \rightarrow G$

SESSION 12

2. In each of the following we give an English sentence and a number of candidate logical expressions written in a suggestive vocabulary. For each of the logical expressions, state whether it (1) correctly expresses the English sentence; (2) is syntactically invalid and therefore meaningless; or (3) is syntactically valid but does not express the meaning of the English sentence.

(a) Paris and Marseilles are both in France.
   i. $In(Paris \land Marseilles, France)$
   ii. $In(Paris, France) \land In(Marseilles, France)$
   iii. $In(Paris, France) \Rightarrow In(Marseilles, France)$

i. 2
ii. 1
iii. 3

noter
when $\exists x$ $\cancel{\rightarrow}$ $\cancel{\land}$
when $\forall x$ $\cancel{\land}$ $\cancel{\rightarrow}$

(b) There is a country that borders both Iraq and Pakistan.
   i. $\exists c.Country(c) \land Borders(c, Iraq) \land Borders(c, Pakistan)$
   ii. $\exists c.Country(c) \rightarrow (Borders(c, Iraq) \land Borders(c, Pakistan))$
   iii. $\exists c.Country(c) \rightarrow (Borders(c, Iraq) \land Borders(c, Pakistan))$
   iv. $\exists c.Borders(Country(c), Iraq) \land Borders(c, Pakistan)$

i. 1
ii. 3
iii. 2
iv. 2

(c) All countries that border Ecuador are in South America.
   i. $\forall c.Country(c) \rightarrow (Borders(c, Ecuador) \rightarrow In(c, SouthAmerica))$
   ii. $\forall c.Country(c) \rightarrow (Borders(c, Ecuador) \rightarrow In(c, SouthAmerica))$
   iii. $\forall c.(Country(c) \rightarrow Borders(c, Ecuador)) \rightarrow In(c, SouthAmerica)$
   iv. $\forall c.Country(c) \land Borders(c, Ecuador) \land In(c, SouthAmerica)$

i. 1
ii. 1
iii. 3
iv 3

3. (a) Translate into good, natural English (no *or* or *not*):
   $\forall x, y.(SpeaksLanguage(x, l) \land SpeaksLanguage(y, l) \rightarrow Understands(x, y) \land Understands(y, x))$
   (b) Explain why the previous sentence is entailed by the sentence
   $\forall x, y.(SpeaksLanguage(x, l) \land SpeaksLanguage(y, l) \rightarrow Understands(x, y))$

a) if two people speak the same language, they understand each other
b) if x understand y then it is implied by SpeaksLanguage

4. Consider a vocabulary with the following symbols:
   - *Occupation(p, o)*: predicate denoting that person *p* has occupation *o*;
   - *Customer(p1, p2)*: predicate denoting that person *p1* is a customer of person *p2*;
   - *Boss(p1, p2)*: predicate denoting that person *p1* is a boss of person *p2*;
   - *Doctor*, *Surgeon*, *Lawyer*, *Actor*: constants denoting occupations;
   - *Emily*, *Joe*: constants denoting people.

   Use these symbols to write the following assertions in first-order logic:
   (a) Emily is either a surgeon or a lawyer.
   (b) Joe is an actor, but he also holds another job.
   (c) All surgeons are doctors.
   (d) Joe does not have a lawyer (i.e., is not a customer of any lawyer).
   (e) Emily has a boss who is a lawyer.
   (f) There exists a lawyer all of whose customers are doctors.
   (g) Every surgeon has a lawyer.

a) $Occ(Emily, Surgeon) \lor Occ(Emily, Lawyer)$
b) $Occ(Joe, Actor) \land (\exists j. Occ(Joe, j) \land j \neq Actor)$
c) $\forall p. Occ(p, Surgeon) \rightarrow Occ(p, Doctor)$
d) $\neg(\exists p. Customer(Joe, p) \land Occ(p, Lawyer))$
e) $\exists p. Boss(p, Emily) \land Occ(p, Lawyer)$
f) $\exists p. Occ(p, Lawyer) \land (\forall c. Customer(c, p) \rightarrow Occ(c, Doctor))$
g) $\forall s. Occ(s, Surgeon) \rightarrow (\exists p. Occ(p, Lawyer) \land Customer(s, p))$

13. Explain what is wrong with the following proposed definition of adjacent squares in the wumpus world.
$$\forall x, y.Adjacent([x, y], [x + 1, y]) \land Adjacent([x, y], [x, y + 1])$$

does not check for out of bounds
missing $(x-1, y)$ and $(x, y-1)$

18. (a) Consider a knowledge base containing just two sentences: $P(a)$ and $P(b)$. Does this knowledge base entail $\forall x.P(x)$? Why/why not?
   (b) Write down a logical sentence such that every world in which it is true contains exactly one object.
   (c) Write down a logical sentence such that every world in which it is true contains exactly two objects.

a) no relation between $P(a)$, $P(b)$ and $P(x)$ for $\forall x$
b) $\forall x \; x = y$
c) $\exists! xyab \; x = y \land a = b \land x \neq a$
$\exists xy \; x \neq y \land (\forall z \; z = x \lor z = y)$

SESSION 14

1. The monkey-and-bananas problem is faced by a monkey in a laboratory with some bananas hanging out of reach from the ceiling. A box is available that will enable the monkey to reach the bananas if he climbs on it. Initially, the monkey is at *A*, the bananas at *B*, and the box at *C*. The monkey and box have height *Low*, but if the monkey climbs onto the box he will have height *High*, the same as the bananas. The actions available to the monkey include *Go* from one place to another, *Push* an object from one place to another, *ClimbUp* onto or *ClimbDown* from an object, and *Grasp* or *Ungrasp* an object. The result of a *Grasp* is that the monkey holds the object if the monkey and object are in the same place at the same height.
   (a) Write down the initial state description.

Initial: AT(monkey, A), Height(Monkey, low)
AT(bananas, B), Height(bananas, heigh)
AT(box, C), Height(box, low)

(b) Write the six action schemas.

Go(x,y)	Push(x,y,b)	ClimbUp(x,b)	ClimbDown(x,b)	Grasp(x,t,h)	UnGrasp(x,t,h)
Pre: $!\exists x,y$	Pre: $!\exists x,y,b$	Pre: $!\exists x,b$	Pre: $!\exists x$	Pre: $!\exists x,t$	Pre: $!\exists x,t$
at(entity, x)	at(M, x)	at(M, x)	at(M, x)	at(M, x)	at(M, x)
height(entity, low)	at(b, x)	at(b, x)	at(b, x)	at(t, x)	height(M, h)
res: at(entity, y)	height(M, low)	height(M, low)	height(M, height)	height(M, h)	res: $\neg$ have(t)
$\neg$at(entity, x)	height(b, low)	height(b, low)	height(b, low)	height(t, h)	$\neg$at(b, x)
	res: at(M, y)	res: height(M, height)	res: height(M, low)	res: have(t)	$\neg$height(b, h)
	at(b, y)			$\neg$at(b, x)	
	$\neg$at(M, x)			$\neg$height(b, h)	
	$\neg$at(b, x)				

if (pushable)? if (climbable)? if (graspable)?

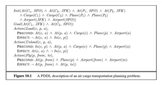


Figure 10.1 A POSL description of an air cargo transportation planning problem.

2. Given the action schemas and initial state from Figure 10.1, what are all the applicable concrete instances of *Fly(p, from, to)* in the state described by the following formula?
$$At(P_1, JFK) \land At(P_2, SFO) \land Plane(P_1) \land Plane(P_2) \land Airport(JFK) \land Airport(SFO)$$

fly(P_1, JFK, JFK)
fly(P_1, JFK, SFO)
fly(P_1, SFO, JFK)
fly(P_2, SFO, SFO)

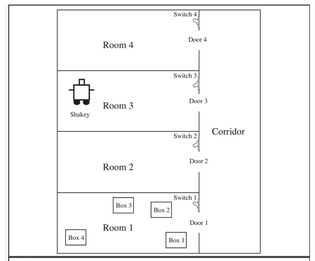


Figure 10.14 Shakey's world. Shakey can move between landmarks within a room, can pass through the door between rooms, can climb climbable objects and push pushable objects, and can flip light switches.

4. The original STRIPS planner was designed to control Shakey the robot. Figure 10.14 shows a version of Shakey's world consisting of four rooms lined up along a corridor, where each room has a door and a light switch. The actions in Shakey's world include moving from place to place, pushing movable objects (such as boxes), climbing onto and down from rigid objects (such as boxes), and turning light switches on and off. The robot itself could not climb on a box or toggle a switch, but the planner was capable of finding and printing out plans that were beyond the robot's abilities. Shakey's six actions are the following:
   1 • *Go*(x, y, r), which requires that Shakey be at x and that x and y be locations in the same room r (expressed by adequate predicates *At* and *In*). By convention a door between two rooms is in both of them.
   2 • Push a box b from location x to location y within the same room: *Push*(b, x, y, r). You will need the predicate *Box* and constants for the boxes.
   3 • Climb onto a box from position x: *ClimbUp*(x, b); climb down from a box to position x: *ClimbDown*(b, x). You will need the predicate *On* and the constant *Floor*.
   4 • Turn a light switch on or off: *TurnOn*(s, b); *TurnOff*(s, b). To turn a light on or off, Shakey must be on top of a box at the light switch's location.

1) (:action Go :parameters (?x ?y ?r)
   :precondition (and (at(shakey ?x)) (in(?x ?r)) (in(?y ?r)) (on shakey floor))
   :effects (and (at(shakey ?y)) (not (at(shakey ?x))))

2) (:action Push :parameters (?b ?x ?y ?r)
   :precondition
   :effects)

3) (:action climbOn :parameters (?b ?x ?y ?r)
   :precondition
   :effects)

3) (:action climbOn :parameters (?b ?x ?y ?r)
   :precondition
   :effects)

4) (:action turnOn :parameters (?b ?x ?y ?r)
   :precondition
   :effects)

4) (:action turnOff :parameters (?b ?x ?y ?r)
   :precondition
   :effects)

Write PDDL sentences for Shakey's six actions and the initial state. From these, construct a plan for Shakey to get Box_2 into $Room_2$.

go(ShakeyStart, Door 3, Room 3)
go(Door 3, Door 1, Corridor)
go(Door 1, Box 2, Room 1)
push(Box2, LocBox2, Door1, Room1)
push(Box2, Door1, Door2, Corridor)