1.A

8.1 A logical knowledge base represents the world using a set of sentences with no explicit structure. An **analogical** representation, on the other hand, has physical structure that corresponds directly to the structure of the thing represented. Consider a road map of your country as an analogical representation of facts about the country—it represents facts with a map language. The two-dimensional structure of the map corresponds to the two-dimensional surface of the area.

a. Give five examples of *symbols* in the map language.

b. An *explicit* sentence is a sentence that the creator of the representation actually writes down. An *implicit* sentence is a sentence that results from explicit sentences because of properties of the analogical representation. Give three examples each of *implicit* and *explicit* sentences in the map language.

a. City markers, road signs, monuments, hospitals and rivers

b. Implicit: Hospital NO.1 is larger than NO.2, rivers A is more breadth than rivers B, the shortest way to monuments is as follows
Explicit: There is a hospital named NO.1, rivers A is straight, monuments located on X area.

B.

8.3 Is the sentence $\exists x, y \;\; x = y$ valid? Explain.

Is valid. Because every model contains at least one domain element, for any model, there is the same x and y. So the sentence is valid.

C.

8.6 Which of the following are valid (necessarily true) sentences?

a. $(\exists x \; x = x) \Rightarrow (\forall y \;\; \exists z \; y = z)$.

b. $\forall x \;\; P(x) \vee \neg P(x)$.

c. $\forall x \;\; Smart(x) \vee (x = x)$.

a. Valid. For right hand sentences, for any y there is a z equal to y, so the RHS is valid. For left hand sentences, x equal to x is always true. Therefore, the whole sentence is valid.

b. Valid. For any x, P(x) ∨ ¬P(x) is true.

c. Valid. For any x, x=x is true so Smart (x) ∨ (x=x) must be true.

D.

8.9 This exercise uses the function $MapColor$ and predicates $In(x, y)$, $Borders(x, y)$, and $Country(x)$, whose arguments are geographical regions, along with constant symbols for various regions. In each of the following we give an English sentence and a number of candidate logical expressions. 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) $In(Paris \wedge Marseilles, France)$.
(ii) $In(Paris, France) \wedge In(Marseilles, France)$.
(iii) $In(Paris, France) \vee In(Marseilles, France)$.

b. There is a country that borders both Iraq and Pakistan.

(i) $\exists c \quad Country(c) \wedge Border(c, Iraq) \wedge Border(c, Pakistan)$.
(ii) $\exists c \quad Country(c) \Rightarrow [Border(c, Iraq) \wedge Border(c, Pakistan)]$.
(iii) $[\exists c \quad Country(c)] \Rightarrow [Border(c, Iraq) \wedge Border(c, Pakistan)]$.
(iv) $\exists c \quad Border(Country(c), Iraq \wedge Pakistan)$.

a. i. (2). Conjunction cannot be used inside a term.
ii. (1)
iii. (3) Disjunction cannot represent “both”

b. i. (1)
ii. (3) implication cannot be used in existential
iii. (2)c used outside its quantifier.
iv. (2)conjunction cannot be used inside a term

E.

8.10 Consider a vocabulary with the following symbols:

$Occupation(p, o)$: Predicate. Person p has occupation o.
$Customer(p1, p2)$: Predicate. Person $p1$ is a customer of person $p2$.
$Boss(p1, p2)$: Predicate. 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.

a. O(E,S) ∨ O(E,L).
b. O(J,A) ∧ ∃p p≠A ∧ O(J, p).

2.F

9.23 From "Horses are animals," it follows that "The head of a horse is the head of an animal." Demonstrate that this inference is valid by carrying out the following steps:

a. Translate the premise and the conclusion into the language of first-order logic. Use three predicates: $HeadOf(h, x)$ (meaning "h is the head of x"), $Horse(x)$, and $Animal(x)$.

b. Negate the conclusion, and convert the premise and the negated conclusion into conjunctive normal form.

a. ∀ x Horse(x) ⇒ Animal(x)

∀ x, h Horse(x) ∧ HeadOf(h, x) ⇒ ∃y Animal(y) ∧ HeadOf(h, y)

b. A. ¬Horse(x) ∨ Animal(x)

B. Horse(G)

C. HeadOf(H,G)

D. ¬Animal(y) ∨ ¬HeadOf(H, y)

G.

9.24 Here are two sentences in the language of first-order logic:

(A) $\forall x \ \exists y \ (x \geq y)$

(B) $\exists y \ \forall x \ (x \geq y)$

a. Assume that the variables range over all the natural numbers $0, 1, 2, \ldots, \infty$ and that the "$\geq$" predicate means "is greater than or equal to." Under this interpretation, translate (A) and (B) into English.

b. Is (A) true under this interpretation?

c. Is (B) true under this interpretation?

a. (A). For every natural number there is some other natural number that is smaller than or equal to it

(B). There is a particular natural number that is smaller than or equal to any natural number.

b. Yes

c. Yes

3.H

10.2 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

$$At(P_1, JFK) \wedge At(P_2, SFO) \wedge Plane(P_1) \wedge Plane(P_2) \wedge Airport(JFK) \wedge Airport(SFO) \ ?$$

Fly(P1, JFK, SFO)

Fly(P1, JFK, JFK)

Fly(P2, SFO, JFK)

Fly(P2, SFO, SFO)

I.

10.3 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.

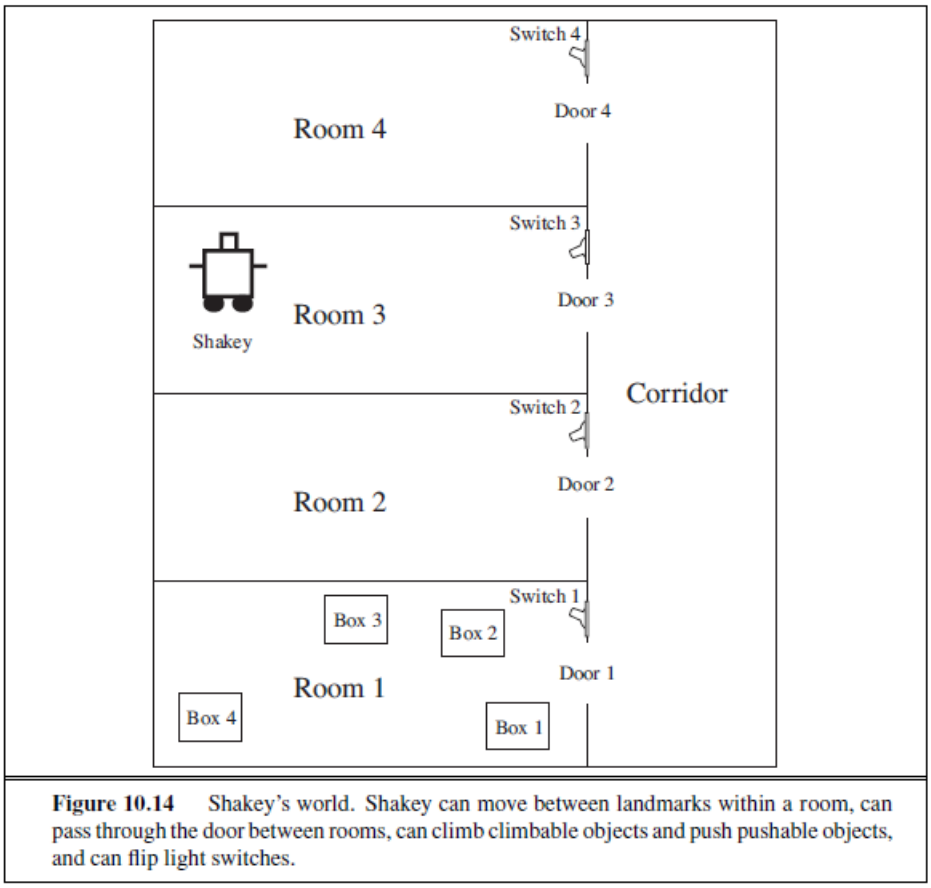


Figure 10.14 Shakey's world. Shakey can move between landmarks within a room, can pass through the door between rooms, can climb climbable objects and push pushable objects, and can flip light switches.

b. Write the six action schemas.

c. Suppose the monkey wants to fool the scientists, who are off to tea, by grabbing the bananas, but leaving the box in its original place. Write this as a general goal (i.e., not assuming that the box is necessarily at C) in the language of situation calculus. Can this goal be solved by a classical planning system?

d. Your schema for pushing is probably incorrect, because if the object is too heavy, its position will remain the same when the $Push$ schema is applied. Fix your action schema to account for heavy objects.

a. At(Monkey,A) ∧ At(Bananas,B) ∧ At(Box,C) ∧ Height(Monkey,Low) ∧ Height(Box,Low) ∧ Height(Bananas,High) ∧ Pushable(Box) ∧ Climbable(Box)

b. Action(ACTION:Go(x,y),
PRECOND:At(Monkey,x),
EFFECT:At(Monkey,y) ∧ ¬(At(Monkey,x)))

Action(ACTION:Push(b, x, y),
PRECOND:At(Monkey,x) ∧ Pushable(b),
EFFECT:At(b, y) ∧ At(Monkey, y) ∧ ¬At(b, x) ∧ ¬At(Monkey,x))

Action(ACTION:ClimbUp(b),
PRECOND:At(Monkey,x) ∧ At(b, x) ∧ Climbable(b),
EFFECT:On(Monkey, b) ∧ ¬Height(Monkey,High))

Action(ACTION:Grasp(b),
PRECOND:Height(Monkey,h) ∧ Height(b, h) ∧ At(Monkey,x) ∧ At(b, x),
EFFECT:Have(Monkey, b))

Action(ACTION:ClimbDown(b),
PRECOND:On(Monkey, b) ∧ Height(Monkey,High),
EFFECT:¬On(Monkey, b) ∧ ¬Height(Monkey,High) ∧ Height(Monkey,Low)

Action(ACTION:UnGrasp(b), PRECOND:Have(Monkey, b), EFFECT:¬Have(Monkey, b))

c. the goal should be: Have(Monkey,Bananas, s) ∧ (∃x At(Box, x, s0) ∧ At(Box, x, s))
But there cannot be any relation between these two states.
So the goal state could not be represented

J.

10.4 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:

- $Go(x, y, r)$, which requires that Shakey be $At\ x$ and that x and y are locations In the same room r. By convention a door between two rooms is in both of them.
- 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.
- Climb onto a box from position x: $ClimbUp(x, b)$; climb down from a box to position x: $ClimbDown(b, x)$. We will need the predicate On and the constant $Floor$.
- 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.

Write PDDL sentences for Shakey's six actions and the initial state from Figure 10.14. Construct a plan for Shakey to get Box_2 into $Room_2$.

Actions:

Action(ACTION:Go(x,y),
PRECOND:At(Shakey,x) ∧ In(x,r) ∧ In(y,r),
EFFECT:At(Shakey,y) ∧ ¬(At(Shakey,x)))

Action(ACTION:Push(b,x,y),
PRECOND:At(Shakey,x) ∧ Pushable(b),
EFFECT:At(b,y) ∧ At(Shakey,y) ∧ ¬At(b,x) ∧ ¬At(Shakey,x))

Action(ACTION:ClimbUp(b),
PRECOND:At(Shakey,x) ∧ At(b,x) ∧ Climbable(b),
EFFECT:On(Shakey,b) ∧ ¬On(Shakey,Floor))

Action(ACTION:ClimbDown(b),
PRECOND:On(Shakey,b),
EFFECT:On(Shakey,Floor) ∧ ¬On(Shakey,b))

Action(ACTION:TurnOn(l),
PRECOND:On(Shakey,b) ∧ At(Shakey,x) ∧ At(l,x),
EFFECT:TurnedOn(l))

Action(ACTION:TurnOff(l),
PRECOND:On(Shakey,b) ∧ At(Shakey,x) ∧ At(l,x),
EFFECT:¬TurnedOn(l))

Initial states:

In(Switch1,Room1) ∧ In(Door1,Room1) ∧ In(Door1,Corridor)
In(Switch1,Room2) ∧ In(Door2,Room2) ∧ In(Door2,Corridor)
In(Switch1,Room3) ∧ In(Door3,Room3) ∧ In(Door3,Corridor)
In(Switch1,Room4) ∧ In(Door4,Room4) ∧ In(Door4,Corridor)
In(Shakey,Room3) ∧ At(Shakey,XS)
In(Box1,Room1) ∧ In(Box2,Room1) ∧ In(Box3,Room1) ∧ In(Box4,Room1)
Climbable(Box1) ∧ Climbable(Box2) ∧ Climbable(Box3) ∧ Climbable(Box4)
Pushable(Box1) ∧ Pushable(Box2) ∧ Pushable(Box3) ∧ Pushable(Box4)
At(Box1,X1) ∧ At(Box2,X2) ∧ At(Box3,X3) ∧ At(Box4,X4)
TurnwdOn(Switch1) ∧ TurnedOn(Switch4)

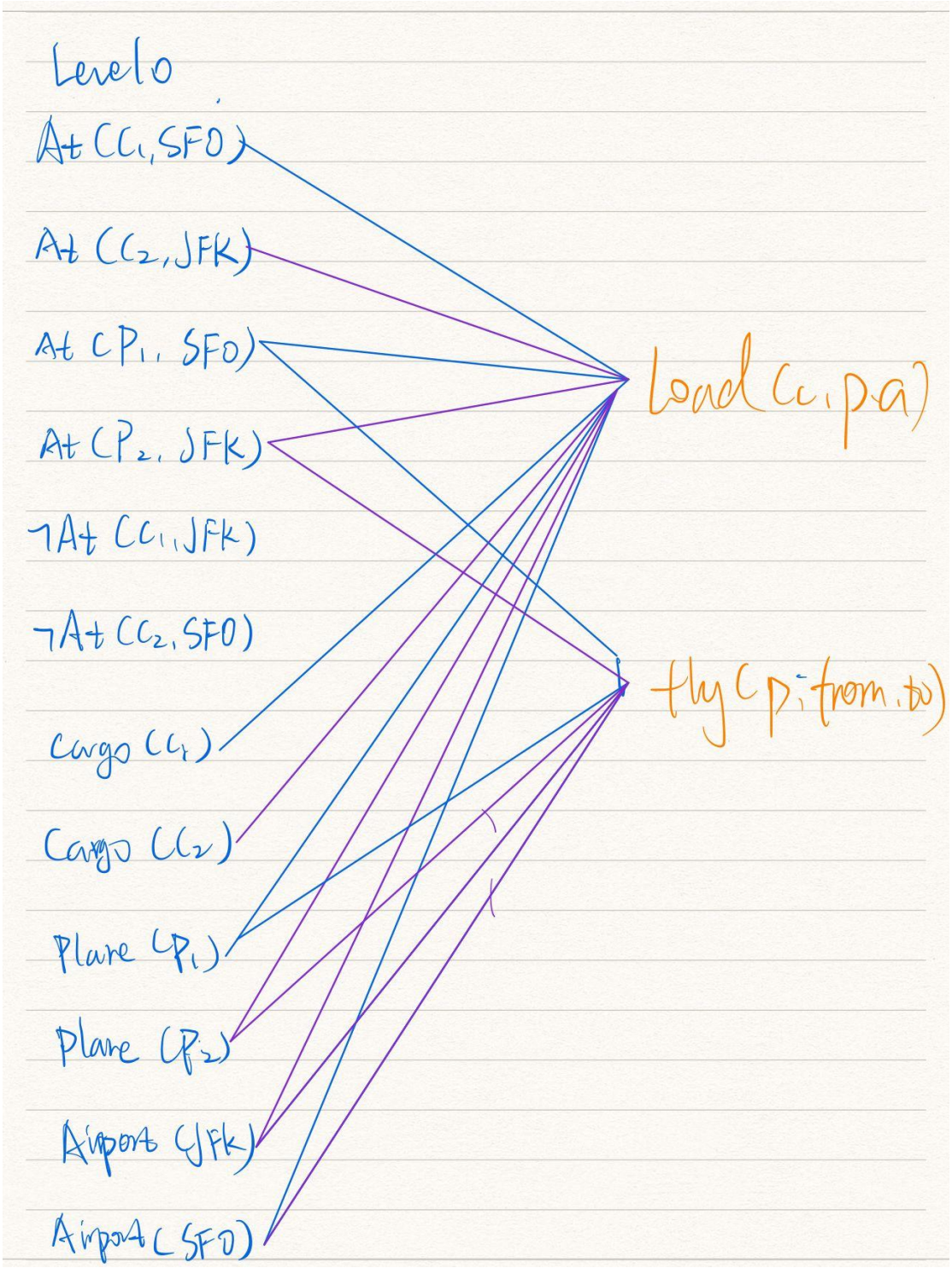
Achieve the goal:
Go(XS,Door3)
Go(Door3,Door1)
Go(Door1,X2)
Push(Box2,X2,Door1)
Push(Box2,Door1,Door2)
Push(Box2,Door2, Switch2)

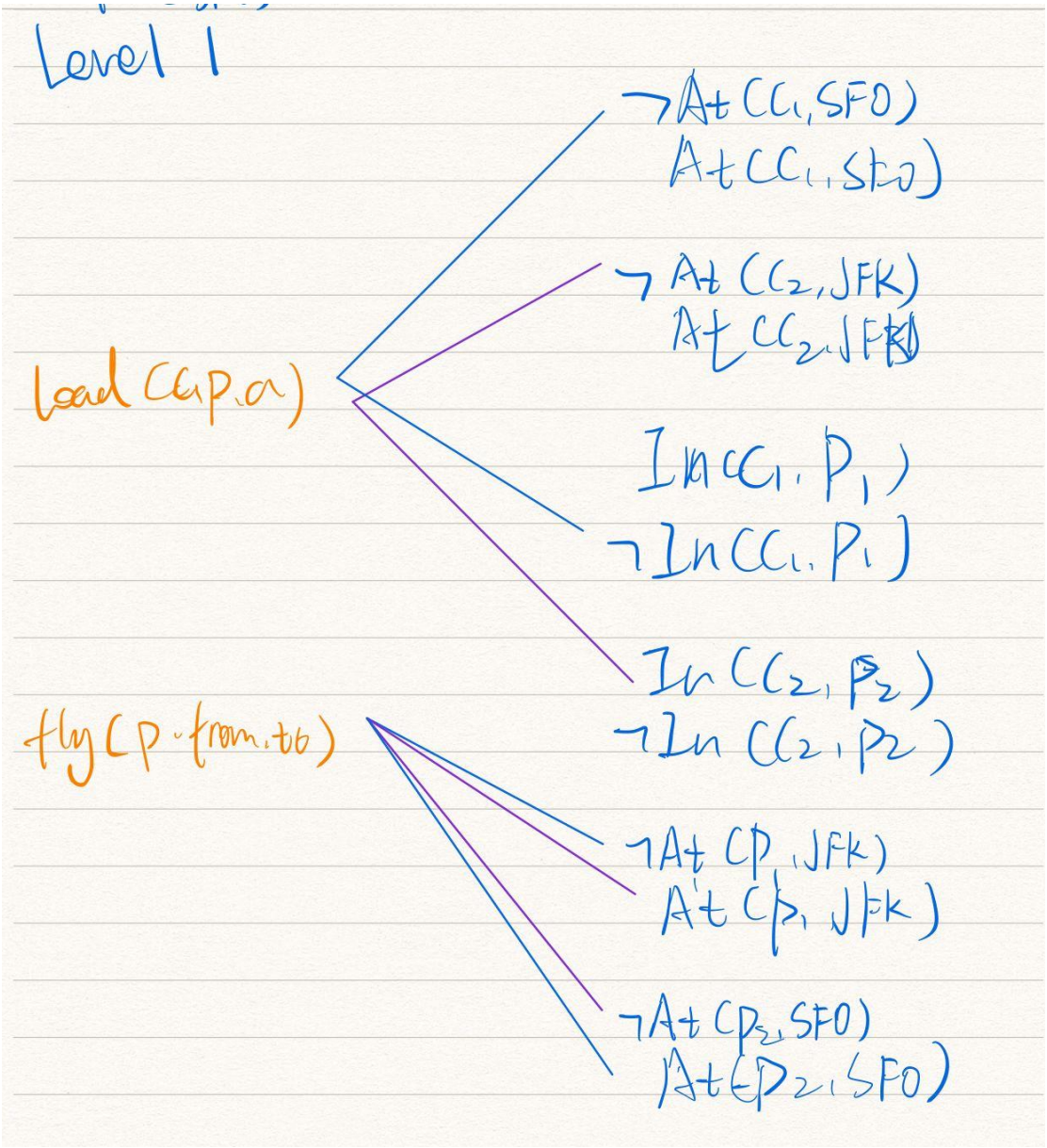
K.

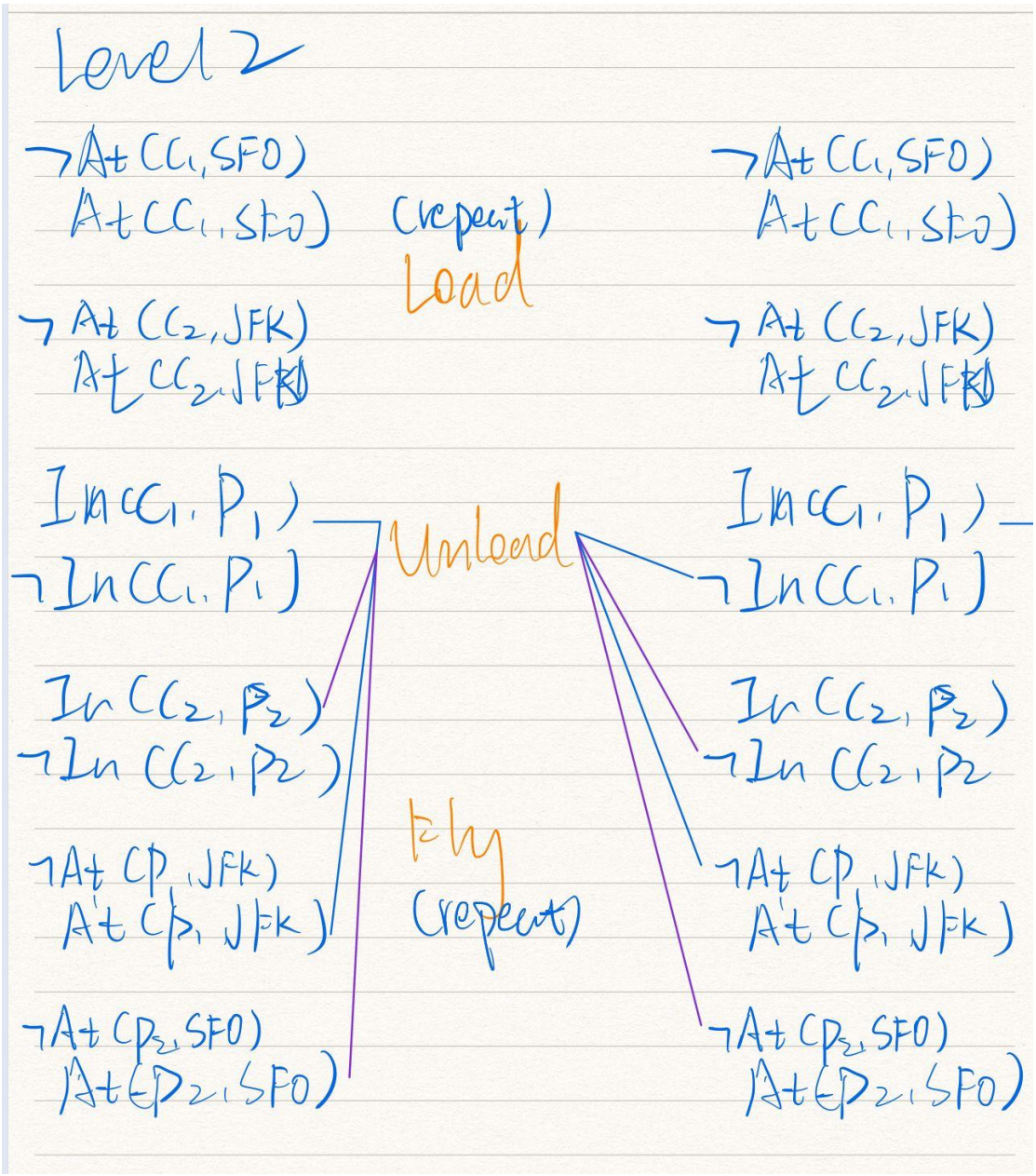
10.9 Construct levels 0, 1, and 2 of the planning graph for the problem in Figure 10.1.

$Init(At(C_1, SFO) \land At(C_2, JFK) \land At(P_1, SFO) \land At(P_2, JFK)$
$\quad \land Cargo(C_1) \land Cargo(C_2) \land Plane(P_1) \land Plane(P_2)$
$\quad \land Airport(JFK) \land Airport(SFO))$
$Goal(At(C_1, JFK) \land At(C_2, SFO))$
$Action(Load(c, p, a),$
$\quad$ PRECOND: $At(c, a) \land At(p, a) \land Cargo(c) \land Plane(p) \land Airport(a)$
$\quad$ EFFECT: $\neg At(c, a) \land In(c, p))$
$Action(Unload(c, p, a),$
$\quad$ PRECOND: $In(c, p) \land At(p, a) \land Cargo(c) \land Plane(p) \land Airport(a)$
$\quad$ EFFECT: $At(c, a) \land \neg In(c, p))$
$Action(Fly(p, from, to),$
$\quad$ PRECOND: $At(p, from) \land Plane(p) \land Airport(from) \land Airport(to)$
$\quad$ EFFECT: $\neg At(p, from) \land At(p, to))$

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


Level 0
At (C1, SFO)
At (C2, JFK)
At (P1, SFO)
At (P2, JFK)
¬At (C1, JFK)
¬At (C2, SFO)
Cargo (C1)
Cargo (C2)
Plane (P1)
Plane (P2)
Airport (JFK)
Airport (SFO)
load (c, p, a)
fly (p, from, to)


Level 1
Load(C1,P,A)
Fly(P,from,to)
¬At(C1,SFO)
At(C1,SFO)
¬At(C2,JFK)
At(C2,JFK)
In(C1,P1)
¬In(C1,P1)
In(C2,P2)
¬In(C2,P2)
¬At(P1,JFK)
At(P1,JFK)
¬At(P2,SFO)
At(P2,SFO)


Level 2
¬At(C1, SFO)
At(C1, SFO)
(repeat)
Load
¬At(C2, JFK)
At(C2, JFK)
In(C1, P1)
¬In(C1, P1)
Unload
In(C2, P2)
¬In(C2, P2)
¬At(P1, JFK)
At(P1, JFK)
Fly
(repeat)
¬At(P2, SFO)
At(P2, SFO)
¬At(C1, SFO)
At(C1, SFO)
¬At(C2, JFK)
At(C2, JFK)
In(C1, P1)
¬In(C1, P1)
In(C2, P2)
¬In(C2, P2)
¬At(P1, JFK)
At(P1, JFK)
¬At(P2, SFO)
At(P2, SFO)