**Q1:**

a)

Possible goal states:

1. At(box1, b4) ^ At(box2, e5)
2. At(box2, b4) ^ At(box1, e5)

b)

Action(MoveLeft(p, y1, y2),

Precond: Position(y1) ^ Position(y2) ^ Player(p) ^ Left(y2, y1) ^ Blank(y2) ^ At(p, y1)

Effect: ~At(p,y1) ^ ~Blank(y2) ^ At(p, y2) ^ Blank(y1)

Action(PushLeft(p, b, y1, y2, y3),

Precond: Position(y1) ^ Position(y2) ^ Position(y3) ^ Player(p) ^ Box(b) ^ Left(y3, y2) ^ Left(y2, y1) ^ Blank(y3) ^ At(p, y1) ^ At(b, y2)

Effect: ~At(p, y1) ^ ~At(b, y2) ^ ~Blank(y3) ^ At(p, y2) ^ At(b, y3) ^ Blank(y1)

c)

l, d, D, l, U, d, d, d, r, U, U, l, u, R, R, u, l, u, l, D, D

**Q2:**

False. Each sequence of primitive actions in a refinement usually describes one parent high-level action. However, the overall task could require a sequence of high-level actions. This means that a specific sequence might not be shown all together, like when half the sequence is in one refinement and the other half is in a different refinement. However, when the high-level actions are implemented as their primitive actions and the plan is formed, the sequence could show itself.

**Q3:**

Option: Repair bike

-$200: 30% chance

-$80: 60% chance

-$400: 10% chance

EU(Repair bike) = 0.3(-200) + 0.6(-80) + 0.1(-400) = -$148

Option: Buy new bike

-$200: 100% chance

EU(Buy new bike) = -200

Assuming no utility of money or certainty equivalent,

since the EU(Repair bike) > EU(Buy new bike)

Alice should get the bike repaired (Option a).