Logics for Safe AI 2024/2025
Coursework 3

Aran Montero 9540318
Pablo Pardos 843586
Jesse Hoiting 4443306

3.1
For the states we just have to take into account all possible permutations of the three cards which would translate to a set S = [(1a,2b,3c), (1a,3b,2c), (2a,1b,3c), (2a,3b,1c), (3a,1b,2c), (3a,2b,1c)]

For the indistinguishability relations we need to define one per agent. For fi being the indistinguishability relation for the ith agent we have that:

f1 = [(1a,2b,3c),(1a,3b,2c)] ,[(2a,1b,3c),(2a,3b,1c)] [(3a,1b,2c),(3a,2b,1c)]
f2 = [(2a,1b,3c),(3a,1b,2c)] ,[(1a,2b,3c),(3a,2b,1c)] [(1a,3b,2c),(2a,3b,1c)]
f1 = [(3a,2b,1c),(2a,3b,1c)] ,[(1a,3b,2c),(3a,1b,2c)] [(1a,2b,3c),(2a,1b,3c)]

We will not explicitly refer to the indistinguishability of states in respect to themselves because of reflexivity.

The valuation function for the propositions would be pretty straight forward:

V(a1)=[(1a,2b,3c),(1a,3b,2c)]
V(a2)=[(2a,1b,3c),(2a,3b,1c)]
V(a3)=[(3a,1b,2c),(3a,2b,1c)]
V(b1)=[(2a,1b,3c),(3a,1b,2c)]
V(b2)=[(1a,2b,3c),(3a,2b,1c)]
V(b3)=[(1a,3b,2c),(2a,3b,1c)]
V(c1)=[(2a,3b,1c),(3a,2b,1c)]
V(c2)=[(1a,3b,2c),(3a,1b,2c)]
V(c3)=[(1a,2b,3c),(2a,1b,3c)]

3.2
Given q123:

a. It would be represented as D{a,b}(c3) which does **not** hold as for it to do so c3 must hold in all states indistinguishable by either of them. Just because a could consider both q123 and q132 to be active (indistinguishable by a) it does not hold anymore.
b. C{a,b} a1 or b2 or c3 does **not** hold because for it to do so everybody must know that everybody knows that everybody knows… etc. This is because for example in the case of b that knows b2 even in the state q321 it would know that from the point of view of a (a3) it would be not be certain that a knows if its true.

c. C{a,b}(c1 ∨ c2 ∨ c3) holds because in every possible state all parts know that c must be holding the remaining card, which could only be one of those three.

3.3

To be able to express “agent a knows that agent b knows which card agent b has” (in ELCD) we might break this expression down to understand it easily:

- First of all, b1,b2 and b3 are the propositions that represent agent b having the card with 1 dot, 2 dots, or 3 dots, respectively.
- Agent b knowledge would be expressed as Kb(b1 ∨ b2 ∨ b3), because agent b knows it has one of the three cards. In any given state of the model M_abc, agent b has exactly one of the three cards, and agent b is aware of which card it holds. Thus, this statement is true in all states of the model.

But we want to express that agent “a” knows that. Then, the expression would be KaKb(b1 ∨ b2 ∨ b3) , which means means that agent a knows that agent b knows which card it holds. The reason is that b always knows which card it has, agent a cannot consider any state possible where agent b does not know which card it has.

Therefore, because agent b always knows which card it holds and the knowledge operator is veridical, the formula KaKb(b1 ∨ b2 ∨ b3) is true in all states of the model M_abc (veridicality of knowledge and reflexivity).

3.4

In this case, we are talking about only one agent’s knowledge (agent “a”). Then, we will only be talking about Ka.

At the same time, propositions will be the same because we are talking about the same information (knowledge): which card agent b has, expressed as (b1 ∨ b2 ∨ b3). If we negate that agent a knows which card agent b has, the formula in ELCD that expresses this is ¬Ka(b1 ∨ b2 ∨ b3).

The Kripke model M_abc represents a scenario where each agent knows only their own card, but not the other agents' cards. In any state of M_abc, there will be multiple states that are indistinguishable for agent a in terms of what cards the other agents have. Agent a will not be able to distinguish the states where agent b has the card with 1 dot from the states where agent b has the card with 2 or 3 dots, and so on. Therefore, agent a does not know which specific card agent b holds.

Then, the formula ¬Ka(b1 ∨ b2 ∨ b3) is true in all states of the model M_abc, given the rules of the card game scenario.

3.5

<u>Local states</u>

The local states of an agent correspond to the cards that agent could have. Then, for each agent:

- The local states for agent a:
    - a1: Agent a has the card with 1 dot.
    - a2: Agent a has the card with 2 dots.
    - a3: Agent a has the card with 3 dots.
- The local states for agent b:
    - b1: Agent b has the card with 1 dot.
    - b2: Agent b has the card with 2 dots.
    - b3: Agent b has the card with 3 dots.
- The local states for agent c:
    - c1: Agent c has the card with 1 dot.
    - c2: Agent c has the card with 2 dots.
    - c3: Agent c has the card with 3 dots.

Environment states

The states of the environment correspond to the possible combinations of cards (that can be dealt to the agents). There are a total of six possible states which we can represent as the global state of the system:

- q123: Agent a has card 1, agent b has card 2, and agent c has card 3.
- q132: Agent a has card 1, agent b has card 3, and agent c has card 2.
- q213: Agent a has card 2, agent b has card 1, and agent c has card 3.
- q231: Agent a has card 2, agent b has card 3, and agent c has card 1.
- q312: Agent a has card 3, agent b has card 1, and agent c has card 2.
- q321: Agent a has card 3, agent b has card 2, and agent c has card 1.

Global states

The global states of the interpreted system are combinations of local states of all agents and the environment state. Then:

- (q123, a1, b2, c3): Environment is “q123”, agent a's local state is "a1", agent b's local state is "b2", agent c's local state is "c3".
- (q132, a1, b3, c2): Environment is “q132”, agent a's local state is "a1", agent b's local state is "b3", agent c's local state is "c2".
- (q213, a2, b1, c3): Environment is “q213”, agent a's local state is "a2", agent b's local state is "b1", agent c's local state is “c3".
- (q231, a2, b3, c1): Environment is “q231”, agent a's local state is "a2", agent b's local state is "b3", agent c's local state is "c1".
- (q312, a3, b1, c2): Environment is “q312”, agent a's local state is "a3", agent b's local state is "b1", agent c's local state is "c2".
- (q321, a3, b2, c1): Environment is “q321”, agent a's local state is "a3", agent b's local state is "b2", agent c's local state is "c1".

These global states capture all the possible combinations of environment states and local states, given that each agent knows its own card.

3.6

There are 4 states:
St= {s1,s2,s3,s4}
Each state is described as, also shown in Figure 1:
s1 : (surface: True, open: False, sunk: False)
s2 : (surface: True, open: True, sunk: False)
s3 : (surface: False, open: False, sunk: False)
s4 : (surface: False, open: True, sunk: True)

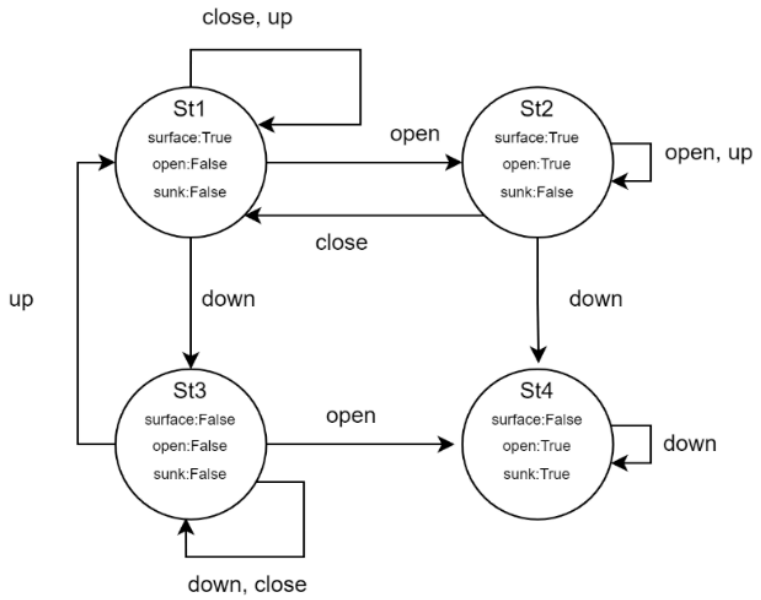

Figure 1: Graphical submarine model

The equivalence classes for the submarine are ~ {s1, s3}, {s2}, {s4}.

3.7

a. It means that the submarine agent either knows that it is sunk or it knows that it has not sunk. For it to hold in all indistinguishable states the submarine must know the state of sunk. by definition it knows this so we can say that it does indeed hold.
b. It is true, as the system would know in the path of open and not sunk that it has to be on the surface for it to happen (otherwise it would be sunk) and viceversa.

3.8 Jesse
See submarine.ispl file.

3.9 Jesse
See submarine.ispl file.