AI – Assignment 1

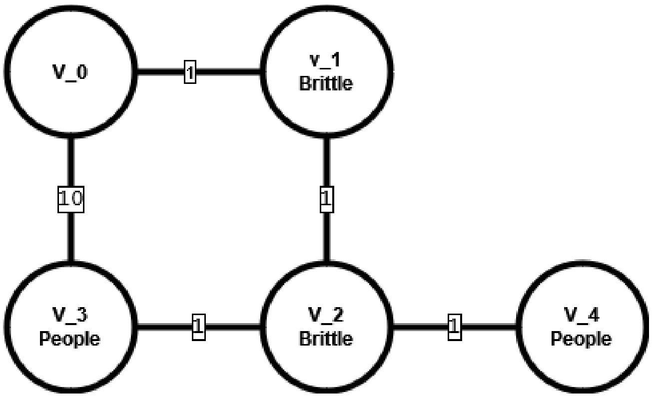
Pan Eyal – 208722058

Question 1

	POKER GAME	HOME REPAIR	COVID-19 RESPONSE	CRYPTARITHMETIC PUZZLES	SOKOBAN PUZZLES
OBSERVABLE	Partially	Yes	Partially	Yes	Yes
DETERMINISTIC	No	Yes	No	Yes	Yes
EPISODIC	No	No	No	No	No
STATIC	Yes	No	No	Yes	Yes
DISCRETE	Yes	No	No	Yes	Yes
SINGLE-AGENT	No	Yes	No	Yes	Yes
AGENT-TYPE	Utility-based	Goal-based	Utility-based	Goal-based	Utility-based

Question 2

Question 6 settings:



Let us represent a state as:

$$S_i: (V, Broken\ V1, Broken\ V2, People\ saved\ at\ V3, People\ saved\ at\ V4)$$

where $V \in \{V_0, V_1, V_2, V_3, V_4\}$, and $People\ saved\ at\ Vi, Broken\ Vi \in \{True, False\}$

The heuristic is: edge weight sum of the minimum spanning tree that containing only vertices that need saving (connecting neighbors of nonessential vertices as necessary, as shown in class).

The initial state is: $S_0: (V_0, F, F, F, F)$ and $g = 0, h = 4, f = 4, A = 12$

For all agents, S_0 is retrieved, and it is not a goal state, so it is expanded with:

$$S_1: (V_1, T, F, F, F), g = 1, h = 3, f = 3, A = 13$$

$$S_2: (V_3, F, F, T, F), g = 10, h = 2, f = 2, A = 2$$

Now, for the agents:

a) Greedy search

Greedy search takes S_2 as it has the lowest f and return with $[V_3]$. In next search it expands with:

$$S_3: (V_2, F, T, T, F), h = f = 1, A = 1$$

Greedy search takes S_3 as it has the lowest f (the state that would have returned to V_0 would cause an entry of duplicate state and therefore ignored) and return with $[V_2]$. In next search it expands with:

$$S_4: (V_1, T, T, T, F), h = f = \infty, A = 2$$

$$S_5: (V_4, F, T, T, T), h = f = 0, A = 0$$

Greedy search takes S_5 as it has the lowest f and return with $[V_4]$. Now it sees that it reached a goal state and terminates.

b) A* search

A* search takes S_1 as it has the lowest $f = 4$ and expands with:

$$S_3: (V_2, T, T, F, F), g = 2, h = 2, f = 4, A = \infty$$

A* search takes S_3 as it has the lowest h (the state that would have returned to V_0 would cause an entry of duplicate state and therefore ignored) and now expands with:

$$S_4: (V_3, T, T, T, F), g = 3, h = 2, f = 5, A = \infty$$

$$S_5: (V_4, T, T, F, T), g = 3, h = 2, f = 5, A = \infty$$

A* takes S_4 as it breaks ties with lower number vertex, and expands with:

$$S_6: (V_0, T, T, T, F), g = 13, h = \infty, f = \infty, A = \infty$$

A* search takes S_5 since it is the lowest $f = 5$, and can't expand since V_2 broke. S_5 is not a goal but it is empty, so A* takes S_2, and expands with:

$$S_7: (V_2, F, T, T, F), g = 11, h = 1, f = 1, A = 1$$

A* search takes S_7 as it has the lowest f (the state that would have returned to V_0 would cause an entry of duplicate state and therefore ignored) and now expands with:

$$S_8: (V_1, T, T, T, F), g = 12, h = 2, f = 14, A = 2$$

$$S_9: (V_4, F, T, T, T), g = 12, h = 0, f = 12, A = 0$$

A* search takes S_9 as it has the lowest f, sees that it is a goal state, returns the sequence: $[V_3, V_2, V_4]$ and terminates.

c) Real-Time A*

RTA* search choose S_1 as it has the lowest $f = 4$ and expands with:

$$S_3: (V_2, T, T, F, F), g = 2, h = 2, f = 4, A = \infty$$

RTA* takes S_3 as it has the lowest h (the state that would have returned to V_0 would cause an entry of duplicate state and therefore ignored) and runs out of allowed expansions. It returns with the sequence: $[V_1, V_2]$. In next search it expands with:

$$S_4: (V_3, T, T, T, F), g = 3, h = 2, f = 5, A = \infty$$

$$S_5: (V_4, T, T, F, T), g = 3, h = 2, f = 5, A = \infty$$

RTA* takes S_4 as it breaks ties with lower number vertex, and expands with:

$$S_6: (V_0, T, T, T, F), g = 13, h = \infty, f = \infty, A = \infty$$

RTA* search takes S_5 since it is the lowest $f = 5$, and runs out of allowed expansions. It returns with the sequence: $[V_4]$. In next search it can't expand since V_2 already broke.
RTA* does not reach the goal!

d) Now with $\boldsymbol{h'(n) = 2 \cdot h(n)}$

It makes no difference for greedy search in this example and in this domain, that is because $f = h'(n)$ won't cause any node queue ordering ($f\ doesn't\ include\ g$).

$S_0: (V_0, F, F, F, F)$ and $g = 0, h' = 2 \cdot h = 8, f = 8, A = 12$

For all agents, S_0 is retrieved, and it is not a goal state, so it is expanded with:

$$S_1: (V_1, T, F, F, F), g = 1, h' = 2 \cdot\ h = 6, f = 3, A = 13$$

$$S_2: (V_3, F, F, T, F), g = 10, h' = 2 \cdot h = 4, f = 2, A = 2$$

It makes no difference for <u>greedy search</u> in this example and in this domain, that is because $f = h'(n)$ won't cause any change in the node queue ordering.

<u>For A*:</u> A* search takes S_1 as it has the lowest $f = 4$ and expands with:

$$S_3: (V_2, T, T, F, F), g = 2, h' = 2 \cdot h = 4, f = 6, A = \infty$$

A* search takes S_3 as it has the lowest h (the state that would have returned to V_0 would cause an entry of duplicate state and therefore ignored) and now expands with:

$$S_4: (V_3, T, T, T, F), g = 3, h' = 2 \cdot h = 4, f = 7, A = \infty$$

$$S_5: (V_4, T, T, F, T), g = 3, h' = 2 \cdot h = 4, f = 7, A = \infty$$

A* takes S_4 as it breaks ties with lower number vertex, and expands with:

$$S_6: (V_0, T, T, T, F), g = 13, h' = 2 \cdot h = \infty, f = \infty, A = \infty$$

A* search takes S_5 since it is the lowest $f = 5$, and can't expand since V_2 broke. S_5 is not a goal but it is empty, so A* takes S_2, and expands with:

$$S_7: (V_2, F, T, T, F), g = 11, h' = 2 \cdot h = 2, f = 13, A = 1$$

A* search takes S_7 as it has the lowest f (the state that would have returned to V_0 would cause an entry of duplicate state and therefore ignored) and now expands with:

$$S_8: (V_1, T, T, T, F), g = 12, h' = 2 \cdot h = 4, f = 16, A = 2$$

$$S_9: (V_4, F, T, T, T), g = 12, h' = 2 \cdot h = 0, f = 12, A = 0$$

A* search takes S_9 as it has the lowest f, sees that it is a goal state,
returns the sequence: $[V_3, V_2, V_4]$ and terminates.

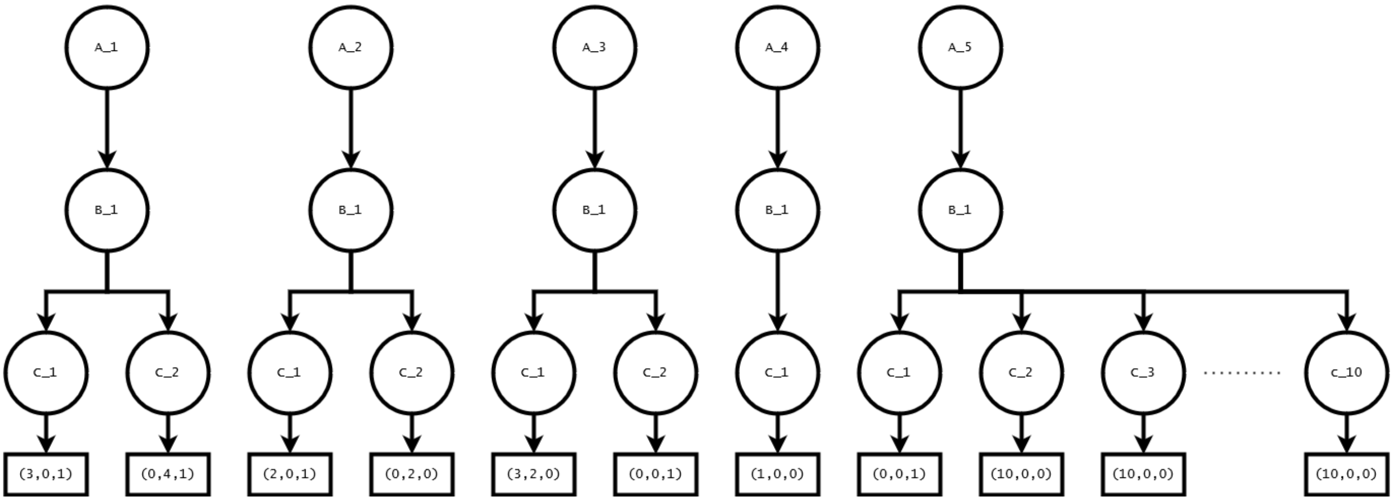
<u>RTA*</u> will behave similarly as A* did.

The multiplication variable wasn't enough to surpass the g part in f and cause the algorithm to choose different path.

h' heuristic is not admissible! We can see in state S_2 for example that $h' = 4$ where $A = 2$.

Question 3

Consider the following tree. The scores represented at the end as: $(A_{score}, B_{score}, C_{score})$



In the case where:

a) **Each agent out for itself, and they cannot communicate:**
In A_1, agent C will choose C_1 as ties broken adversarially (he will prefer one player with score 3, instead of one player with score 4).
That means agent A will receive <u>a score of 3</u>.
In all other choices:
$A_1 > A_2$ because in A_2 max score of agent A is 2
$A_1 > A_3$ because in A_3 agent C will maximize his own score and choose C_2.
$A_1 > A_4$ because in A_4 max score of agent A is 1
$A_1 > A_5$ because in A_5 agent C will maximize his own score and choose C_1.

b) **As in a, except B and C are semi-cooperative:**
In A_2, agent C will choose C_1 as it is prioritizing itself.
That means agent A will receive <u>a score of 2</u>.
In all other choices:
$A_2 > A_1$ because in A_1 agent C will break ties with coop and choose C_2.
$A_2 > A_3$ because in A_3 agent C will maximize his own score and choose C_2.
$A_2 > A_4$ because in A_4 max score of agent A is 1
$A_2 > A_5$ because in A_5 agent C will maximize his own score and choose C_1.

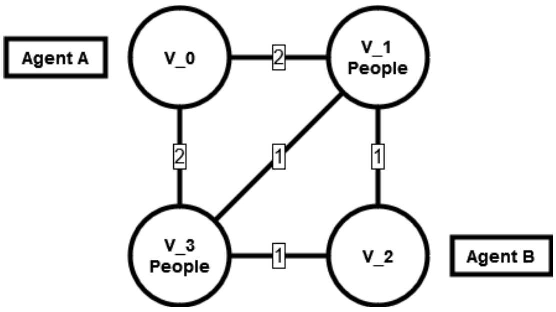
c) **As in a, except B and C are partners aiming to maximize the sum of their scores:**
In A_3, agent C will choose C_1 as it is prioritizing the sum of itself and agent B.
That means agent A will receive <u>a score of 3</u>.
In all other choices:
$A_3 > A_1$ because in A_1 agent C will maximize his and agent B scores and choose C_2.
$A_3 > A_2$ because in A_2 agent C will maximize his and agent B scores and choose C_2.
$A_3 > A_4$ because in A_4 max score of agent A is 1
$A_3 > A_5$ because in A_5 agent C will maximize his and agent B scores and choose C_2.

d) **Paranoid assumption: B and C are against A, no matter what their score is:**
In A_4, agent A will receive <u>a score of 1</u>.
In all other choices agent A can receive a score of 0, and agent C will choose that option.

e) **As in a, except C plays randomly with uniform distribution:**
In A_5, agent A will receive an expected score of $\mathbb{E}[\text{score}(A)] = 0 \cdot 0.1 + (10 \cdot 0.1) \cdot 9 = 9$
In all other choices for agent A, his score will be lower than 9.

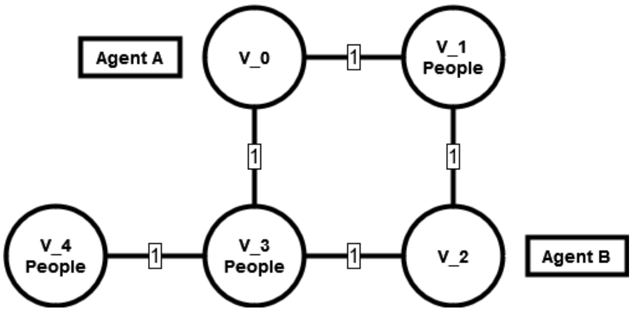
Question 4

a) Example settings: Our graph contains 4 vertices with edges as follows:



Also, we have agents A that starts in V_0 and agent B that starts in V_2, with A moving first.
Agent A starts in vertex V_0 and agent B starts in vertex V_2.
We can easily see that if agent A starts moving towards V_1 or V_3, agent B will be able to move to the same location, reach there faster and save the people before agent A. and then also reach the people on the other vertex as well. In this scenario agent A remains with a <u>score of 0</u>.
If agent A starts with *no_op* action, Agent B will move towards one of the vertices, and then agent A will be able to move to the other location, race agent B, and get a score as well. In this scenario agent A gains a <u>score of 1</u> and ties with agent B.

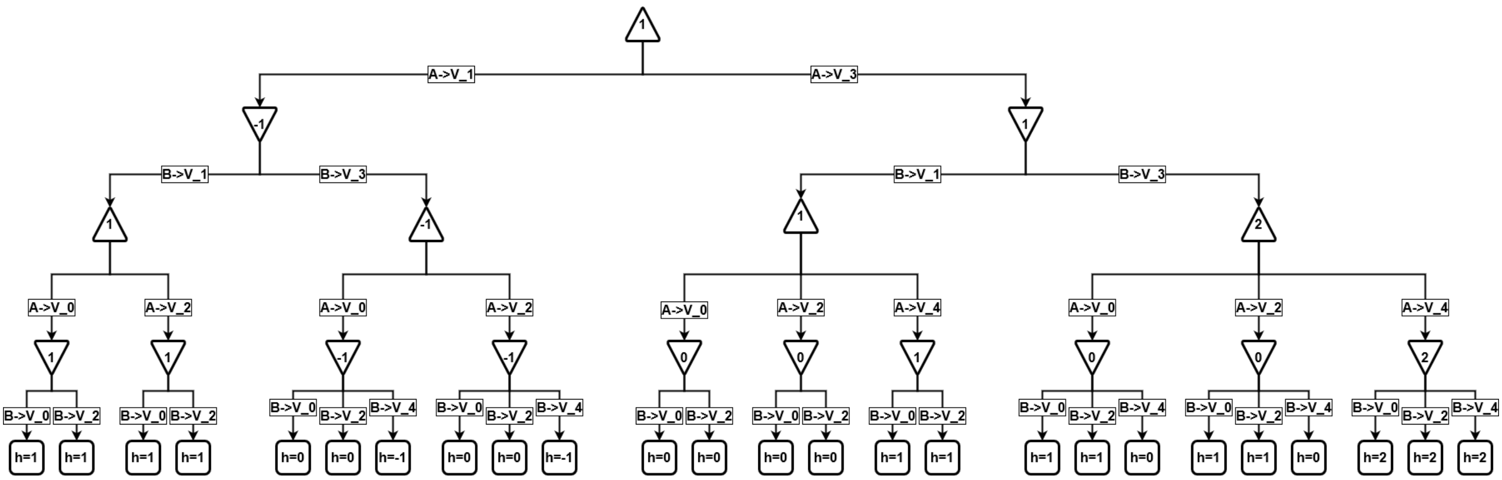
b) Example settings: Our graph contains 5 vertices with edges as follows:



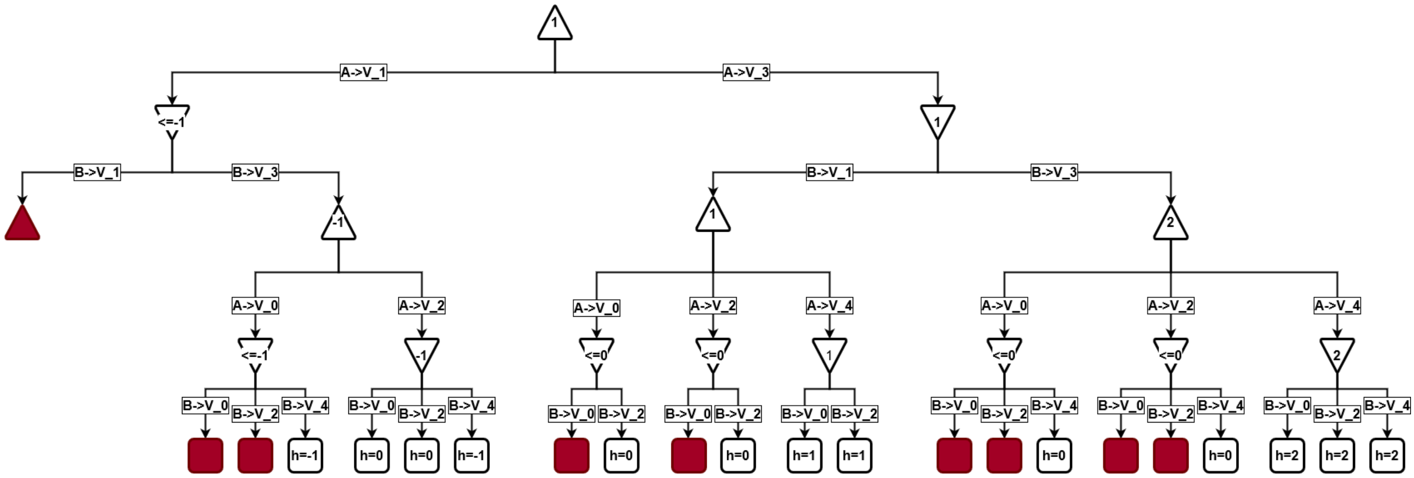
Also, we have agents A that starts in V_0 and agent B that starts in V_2, with A moving first.
The static heuristic function will be the score achieved until current state:

$$h(state) = score(Agent\ 1) - score(Agent\ 2)$$

agent A want to maximize it and agent B wants to minimize it.
We will perform minimax on this graph and receive the following tree for agent A:



c) Now for alpha-beta pruning, red slots in graph mean pruning had occurred. We use the assumption that we evaluate from right to the left (just because more pruning will happen):



Question 5

a) $(\neg A \land \neg B \land \neg C \land \neg D \land \neg E) \lor (A \land B \land C \land D)$
Satisfiable $\rightarrow$ 3 models:

- A through D are true and E is true.
- A through D are true and E is false.
- A through E are false.

b) $(A \lor B \lor C \lor D \lor E \lor F) \land (\neg A \lor \neg B \lor \neg C \lor \neg D \lor \neg E \lor \neg F)$
Satisfiable $\rightarrow 2^6 - 2 = 62$ models:

- All assignments except either all propositions true or all propositions false.

c) $\big(A \lor B \land (D \lor \neg A) \land (E \lor A)\big) \rightarrow \big(B \lor C \land (\neg D \lor E)\big) \land (A \land \neg A)$
Unsatisfiable $\rightarrow$ 0 models:

- Unsatisfiable, because we have a conjunction with the unsatisfiable (A and not A).

d) $\big(A \land (A \rightarrow B) \land (B \rightarrow C) \land (C \rightarrow D)\big) \rightarrow D$
Satisfiable $\rightarrow 2^4 = 16$ models:

- B is entailed by $(A \rightarrow B)$ and C is entailed by $(B \rightarrow C)$ and D entailed by $(C \rightarrow D)$.

e) $\neg\Big(\big(A \land (A \rightarrow B) \land (B \rightarrow C)\big) \rightarrow C\Big)$
Unsatisfiable, because this is a negation of a valid sentence.

f) $(A \rightarrow \neg A) \lor (\neg B \rightarrow B)$
Satisfiable $\rightarrow\ 2^2 - 1 = 3$ models:

- A is false, B is true.
- A is false, B is false.
- A is true, B is true.

Question 6

a) Predicates:

$Vetrex(V0), Vetrex(V1), Vertex(V2), Vetrex(V3), Vertex(V4)$

$Edge(E1, V0, V1), Edge(E2, V0, V3), Edge(E3, V1, V2), Edge(E4, V2, V3), Edge(E5, V2, V4)$

$Weight(E1,1), Weight(E2,10), Weight(E3,1), Weight(E4,1), Weight(E5,1)$

$B(V1), B(V2)$

We will also define edges and their weights for no-op operations:

$Edge(E6, V0, V0), Edge(E7, V1, V1), Edge(E8, V2, V2), Edge(E9, V3, V3), Edge(E10, V4, V4)$

$Weight(E6,1), Weight(E7,1), Weight(E8,1), Weight(E9,1), Weight(E10,1)$

And define trivial inequalities:

$V0 \neq V1 \neq V2 \neq V3 \neq V4$

$E1 \neq E2 \neq E3 \neq E4 \neq E5$

In this world, we are dealing with integers as constants, eg. 1,2,3. To represent calculations based on these constants, we'll define $+, \leq$ predicates to incorporate all the options for "integer" constants in our KB. For example, $+(1,2,3), \leq (1,2), \leq (3,3)$.

Initial fluents:

Loc(V0, S0)

U(V1, S0), U(V2, S0) and auxiliary: U(V0, S0), U(V3, S0) U(V4, S0)

PeopleAt(V3, S0), PeopleAt(V4, S0)

Time(0, S0)

Frame axioms:

1) 1) Equal to themselves

$$\forall v: \, = (v, v)$$

2) Not equal with each other

Create for every $i \in \{1,2,3,4\}$ and $j \in \{1,2,3,4\} \setminus i$ rule: $\neg = (vi, vj)$

2) Undirected graph

$$\forall e, v1, v2: Edge(e, v1, v2) \rightarrow Edge(e, v2, v1)$$

3) Can't be in two places at once

$$\forall s, v1, v2: Loc(v1, s) \wedge Loc(v2, s) \rightarrow = (v1, v2)$$

4) People saved only when visiting their vertex

$$\forall s, v, a: PeopleAt(v, s) \wedge \neg Loc\big(v, Result(a, s)\big) \rightarrow PeopleAt(v, Result(a, s))$$

5) Vertices break only when visiting their vertex

$$\forall s, v, a: B(v) \wedge U(v, s) \wedge \neg Loc\big(v, Result(a, s)\big) \rightarrow U\big(v, Result(a, s)\big)$$

6) Non-brittle vertices do not break

$$\forall s, v, a: \neg B(v) \rightarrow U(v, s)$$

7) Broken vertices stay broken

$$\forall s, v, a: \neg U(v, s) \rightarrow \neg U(v, Result(a, s))$$

Effect axioms:

8) Legal traversal

$$\forall s, e, v1, v2: Loc(v1, s) \wedge Edge(e, v1, v2) \wedge U(v2, s) \rightarrow LT(traverse(e), v1, v2, s)$$

9) Traversal generate a legal location

$$\forall s, e, v1, v2: LT(traverse(e), v1, v2, s) \rightarrow Loc\big(v2, Result(traverse(e), s)\big)$$

10) Saving people:

$$\forall s, e, v1, v2: LT(traverse(e), v1, v2, s) \wedge PeopleAt(v2, s) \\ \rightarrow \neg PeopleAt\big(v2, Result(traverse(e), s)\big)$$

11) Breaking vertices:

$$\forall s, e, v1, v2: LT(traverse(e), v1, v2, s) \wedge B(v2) \rightarrow \neg U\big(v2, Result(tranverse(e), s)\big)$$

b) <u>Part A – conversion to CNF</u>

1) 1) Equal to themselves

$$= (v, v)$$

2) Not equal with each other
Create for every $i \in \{1,2,3,4\}$ and $j \in \{1,2,3,4\} \backslash i$ rule: $\neg = (vi, vj)$

2) Undirected graph

$$\neg Edge(e, v1, v2) \vee Edge(e, v2, v1)$$

3) Can't be in two places at once

$$\neg Loc(v1, s) \vee \neg Loc(v2, s) \vee \neg = (v1, v2)$$

4) People saved only when visiting their vertex

$$\neg PeopleAt(v, s) \vee Loc\big(v, Result(a, s)\big) \vee PeopleAt(v, Result(a, s))$$

5) Vertices break only when visiting their vertex

$$\neg B(v) \vee \neg U(v, s) \vee Loc\big(v, Result(a, s)\big) \vee U\big(v, Result(a, s)\big)$$

6) Non-brittle vertices do not break

$$B(v) \vee U(v, s)$$

7) Broken vertices stay broken

$$U(v, s) \vee \neg U(v, Result(a, s))$$

8) Legal traversal

$$\neg Loc(v1, s) \vee \neg Edge(e, v1, v2) \vee \neg U(v2, s) \vee LT(traverse(e), v1, v2, s)$$

9) Traversal generate a legal location

$$\neg LT(traverse(e), v1, v2, s) \vee Loc\big(v2, Result(traverse(e), s)\big)$$

10) Saving people:

$$\neg LT(traverse(e), v1, v2, s) \vee \neg PeopleAt(v2, s) \\ \vee \neg PeopleAt\big(v2, Result(traverse(e), s)\big)$$

11) Breaking vertices:

$$\neg LT(traverse(e), v1, v2, s) \wedge \neg B(v2) \vee \neg U\big(v2, Result(tranverse(e), s)\big)$$

12) $Vetrex(V0)$
13) $Vetrex(V1)$
14) $Vertex(V2)$
15) $Vetrex(V3)$
16) $Vertex(V4)$
17) $Edge(E1, V0, V1)$
18) $Edge(E2, V0, V3)$
19) $Edge(E3, V1, V2)$
20) $Edge(E4, V2, V3)$
21) $Edge(E5, V2, V4)$
22) $\neg B(V0)$
23) $B(V1)$
24) $B(V2)$
25) $\neg B(V3)$
26) $\neg B(V4)$
27) $U(V0, S0)$

28) $U(V1, S0)$
29) $U(V2, S0)$
30) $U(V3, S0)$
31) $U(V4, S0)$
32) $Loc(V0, S0)$
33) $PeopleAt(V3, S0)$
34) $PeopleAt(V4, S0)$

Part B – Formally defining the query:

Query:

$$\neg PeopleAt(V3, Result(traverse(E2), S0))$$
$$\wedge \neg PeopleAt\left(V4, Result\left(traverse(E5), Result\big(traverse(E4), Result(traverse(E2), S0)\big)\right)\right)$$

Let split this query in two and try to prove each in it's turn:

$$Q1: \neg PeopleAt\big(V3, Result(traverse(E2), S0)\big)$$

$$Q2: \neg PeopleAt\left(V4, Result\left(traverse(E5), Result\big(traverse(E4), Result(traverse(E2), S0)\big)\right)\right)$$

Negation:

$$\neg Q1: PeopleAt\big(V3, Result(traverse(E2), S0)\big)$$

$$\neg Q2: \neg PeopleAt\left(V4, Result\left(traverse(E5), Result\big(traverse(E4), Result(traverse(E2), S0)\big)\right)\right)$$

Part C – Using resolutions to reach a contradiction:

Proving $Q1$

Res: $\neg Q1, (10) \quad \theta = \{v2|V3, e|E2, s|S0\}$

35) $\neg LT(traverse(E2), v1, V3, S0) \vee \neg PeopleAt(V3, S0)$

Res: $(35), (33) \quad \theta = \{\ \ \}$

36) $\neg LT(traverse(E2), v1, V3, S0)$

Res: $(8), (32) \quad \theta = \{v1|V0, s|S0\}$

37) $\neg Edge(e, V0, v2) \vee \neg U(v2, S0) \vee LT(traverse(e), V0, v2, S0)$

Res: $(37), (18) \quad \theta = \{e|E2, v2|V3\}$

38) $\neg U(V3, S0) \vee LT(traverse(E2), V0, V3, S0)$

Res: $(36), (38) \quad \theta = \{v1|V0\}$

39) $\neg U(V3, S0)$

Res: $(39), (30) \quad \theta = \{\ \ \}$

40) $< empty\ clause >$

Proving $Q2$

Res: $(\neg Q2), (10)$ $\theta = \{v2|V4, e|E5, s|Result(traverse(E4), Result(traverse(E2), S0))\}$

35) $\neg LT\left(traverse(E5), v1, V4, Result(traverse(E4), Result(traverse(E2), S0))\right) \vee$
$\neg PeopleAt\left(V4, Result(traverse(E4), Result(traverse(E2), S0))\right)$

Res: $(8), (32)$ $\theta = \{v1|V0, s|S0\}$

36) $\neg Edge(e, V0, v2) \vee \neg U(v2, S0) \vee LT(traverse(e), V0, v2, S0)$

Res: $(36), (18)$ $\theta = \{e|E2, v2|V3\}$

37) $\neg U(V3, S0) \vee LT(traverse(E2), V0, V3, S0)$

Res: $(37), (30)$ $\theta = \{\quad\}$

38) $LT(traverse(E2), V0, V3, S0)$

Res: $(9), (38)$ $\theta = \{e|E2, v1|V0, v2|V3, s|S0\}$

39) $Loc(V3, Result(traverse(E2), S0))$

Res: $(8), (39)$ $\theta = \{v1|V3, s|Result(traverse(E2), S0)\}$

40) $\neg Edge(e, V3, v2) \vee \neg U(v2, Result(traverse(E2), S0)) \vee$
$LT(traverse(e), V3, v2, Result(traverse(E2), S0))$

Res: $(2), (20)$ $\theta = \{e|E4, v1|V2, v2|V3\}$

41) $Edge(E4, V2, V3)$

Res: $(40), (41)$ $\theta = \{e|E4, v2|V2\}$

42) $\neg U(V2, Result(traverse(E2), S0)) \vee$
$LT(traverse(E4), V3, V2, Result(traverse(E2), S0))$

Res: $(5), (24)$ $\theta = \{v|V2\}$

43) $\neg U(V2, s) \vee Loc(V2, Result(a, s)) \vee U(V2, Result(a, s))$

Res: $(43), (29)$ $\theta = \{s|S0\}$

44) $Loc(V2, Result(a, S0)) \vee U(V2, Result(a, S0))$

Res: $(42), (44)$ $\theta = \{a|traverse(E2)\}$

45) $Loc(V2, Result(traverse(E2), S0)) \vee$
$LT(traverse(E4), V3, V2, Result(traverse(E2), S0))$

Res: $(45), (3)$ $\theta = \{v1|V2, s|Result(traverse(E2), S0)\}$

46) $\neg Loc(v2, Result(traverse(E2), S0)) \vee \neg = (V2, v2) \vee$
$LT(traverse(E4), V3, V2, Result(traverse(E2), S0))$

Res: $(46),(39)\quad \theta=\{v2|V2\}$

47) $\neg=(V2,V2)\vee LT\big(traverse(E4),V3,V2,Result(traverse(E2),S0)\big)$

Res: $(47),(1.1)\quad \theta=\{v|V2\}$

48) $LT\big(traverse(E4),V3,V2,Result(traverse(E2),S0)\big)$

Res: $(9),(48)\quad \theta=\{e|E4,v1|V3,v2|V2,s|Result(traverse(E2),S0)\}$

49) $Loc\left(V2,Result\big(traverse(E4),Result(traverse(E2),S0)\big)\right)$

Res: $(8),(49)\quad \theta=\left\{v1|V2,s|Result\big(traverse(E4),Result(traverse(E2),S0)\big)\right\}$

50) $\neg Edge(e,V2,v2)\vee\neg U\left(v2,Result\big(traverse(E4),Result(traverse(E2),S0)\big)\right)\vee$
$LT\left(traverse(e),V2,v2,Result\big(traverse(E4),Result(traverse(E2),S0)\big)\right)$

Res: $(50),(21)\quad \theta=\{e|E5,v2|V4\}$

51) $\neg U\left(V4,Result\big(traverse(E4),Result(traverse(E2),S0)\big)\right)\vee$
$LT\left(traverse(E4),V2,V4,Result\big(traverse(E4),Result(traverse(E2),S0)\big)\right)$

Res: $(6),(51)\quad \theta=\left\{v|V4,s|Result\big(traverse(E4),Result(traverse(E2),S0)\big)\right\}$

52) $B(v)\vee LT\left(traverse(E4),V2,V4,Result\big(traverse(E4),Result(traverse(E2),S0)\big)\right)$

Res: $(52),(26)\quad \theta=\{v|V4\}$

53) $LT\left(traverse(E4),V2,V4,Result\big(traverse(E4),Result(traverse(E2),S0)\big)\right)$

Res: $(35),(53)\quad \theta=\{v1|V4\}$

54) $\neg PeopleAt\left(V4,Result\big(traverse(E4),Result(traverse(E2),S0)\big)\right)$

Res: $(4),(54)\quad \theta=\{v|V4,a|traverse(E4),s|Result(traverse(E2),S0)\}$

55) $\neg PeopleAt\big(V4,Result(traverse(E2),S0)\big)\vee$
$Loc\left(V4,Result\big(traverse(E4),Result(traverse(E2),S0)\big)\right)$

Res: $(3),(55)\quad \theta=\left\{v1|v4,s|,Result\big(traverse(E4),Result(traverse(E2),S0)\big)\right\}$

56) $\neg Loc\left(v2,Result\big(traverse(E4),Result(traverse(E2),S0)\big)\right)\vee\neg=(V4,v2)\vee$
$\neg PeopleAt\big(V4,Result(traverse(E2),S0)\big)$

Res: $(56),(49)\quad \theta=\{v2|V2\}$

57) $\neg=(V4,V2)\vee\neg PeopleAt\big(V4,Result(traverse(E2),S0)\big)$

Res: $(57),(1.1)\quad \theta=\{\quad\}$

58) $\neg PeopleAt\big(V4,Result(traverse(E2),S0)\big)$

Res: $(4),(58)\quad \theta=\{v|V4,a|traverse(E2),s|S0\}$

59) $\neg PeopleAt(V4, S0) \vee Loc\big(V4, Result(traverse(E2), S0)\big)$

Res: $(59), (34) \quad \theta = \{ \quad \}$

60) $Loc\big(V4, Result(traverse(E2), S0)\big)$

Res: $(60), (3) \quad \theta = \{v2|V4, s|Result(traverse(E2), S0)\}$

61) $\neg Loc(v1, Result(traverse(E2), S0)) \vee = (v1, V4)$

Res: $(61), (39) \quad \theta = \{v1|V3\}$

62) $= (V3, V4)$

Res: $(62), (1.2) \quad \theta = \{ \quad \}$

63) $< empty\ clause >$

c) We saw that we just couldn't complete the proof without frame axioms during the resolution.

d) To be able to prove in forward chaining, we must use only Horn Form. We use negations in the knowledge base so forward chaining is not possible.