1 Predicate logic (4 pts)

1.1 Task

Translate the following natural language sentences into formulas of predicate logic:

a) *Every student visiting the Oktoberfest visits the Wasen as well.*

b) *However, not every student visiting the Wasen visits the Oktoberfest.*

c) *No student older than 21 visiting the Wasen remains sober.*

Use the following predicates/functions:

- $\text{IsStudent}(x)$: $\top$ if x is student; $\bot$ otherwise
- $\text{Visits}(x, f); f \in \{\textit{Wasen}, \textit{Oktoberfest}\}$: $\top$ if x visits f; $\bot$ otherwise
- $\text{IsGreaterThan}(x, y); x, y \in \mathbb{R}$: $\top$ if $x > y$; $\bot$ otherwise
- $\text{age}(x)$: returns x's age in years
- $\text{IsSober}(x)$: $\top$ if x is sober; $\bot$ otherwise

1.2 Solution

a) $\text{IsStudent}(x) \wedge \text{Visits}(x, \textit{Oktoberfest}) \rightarrow \text{Visits}(x, \textit{Wasen})$

b) $\neg\forall x(\text{IsStudent}(x) \wedge \text{Visits}(x, \textit{Wasen}) \rightarrow \text{Visits}(x, \textit{Oktoberfest}))$, or
$\exists x(\text{IsStudent}(x) \wedge \text{Visits}(x, \textit{Wasen}) \wedge \neg\text{Visits}(x, \textit{Oktoberfest}))$

c) $\neg\exists x(\text{IsStudent}(x) \wedge \text{IsGreaterThan}(\text{age}(x), 21) \wedge \text{Visits}(x, \textit{Wasen}) \wedge \text{IsSober}(x))$, or
$\text{IsStudent}(x) \wedge \text{IsGreaterThan}(\text{age}(x), 21) \wedge \text{Visits}(x, \textit{Wasen}) \rightarrow \neg\text{IsSober}(x)$

2 Resolution (9pts)

2.1 Task

Assume the following knowledge base:

(a) $C \leftrightarrow A \wedge \neg C$

(b) $\neg D \rightarrow D$

(c) $A \vee \neg D \rightarrow \neg C$

Further, assume the following formula

(d) $A \vee \neg D$

Apply the resolution algorithm to prove or disprove (d).

Which of the following statements is true:

1) (d) can be proven given the knowledge base.
2) (d) can be disproven given the knowledge base.
3) (d) can neither be proven nor disproven given the knowledge base.

2.2 Solution

Let us attempt to prove (d)'s negation. That is, we have to conjunctively combine all the sentences in our knowledge base ((a) to (c)) and (d) and derive the result's CNF:

A	C	D	(a)	(b)	(c)	(d)	(a)∧(b)∧(c)∧(d)	clause	ID
0	0	0	1	0	1	1	0	$A \vee B \vee C$	(e)
0	0	1	1	1	1	0	0	$A \vee B \vee \neg C$	(f)
0	1	0	0	0	0	1	0	$A \vee \neg B \vee C$	(g)
0	1	1	0	1	1	0	0	$A \vee \neg B \vee \neg C$	(h)
1	0	0	0	0	1	1	0	$\neg A \vee B \vee C$	(i)
1	0	1	0	1	1	1	0	$\neg A \vee B \vee \neg C$	(j)
1	1	0	0	0	0	1	0	$\neg A \vee \neg B \vee C$	(k)
1	1	1	0	1	0	1	0	$\neg A \vee \neg B \vee \neg C$	(l)

Now, the algorithm iteratively applies the resolution rule to combinations of clauses, e.g.

- to (e) and (f) resulting in $A \vee B$ (m)
- to (g) and (h) resulting in $A \vee \neg B$ (n)
- to (i) and (j) resulting in $\neg A \vee B$ (o)

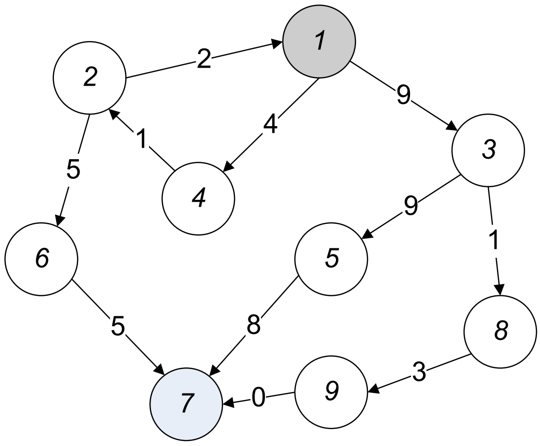
- to (k) and (l) resulting in $\neg A \vee \neg B$ (p)
- to (m) and (n) resulting in A (q)
- to (o) and (p) resulting in $\neg A$ (r)
- to (q) and (r) resulting in $\perp$ (s)

(s) proves contradiction and, therefore, that (d)'s negation holds true, i.e., (d) could be *disproven*. Attempting the same steps with (d)'s negation rather than (d) itself produces no solution, i.e., there is no contradiction in the knowledge base to begin with. Accordingly, statement 2) is true.

3 A* (9pts)

3.1 Task

We are given the following graph:



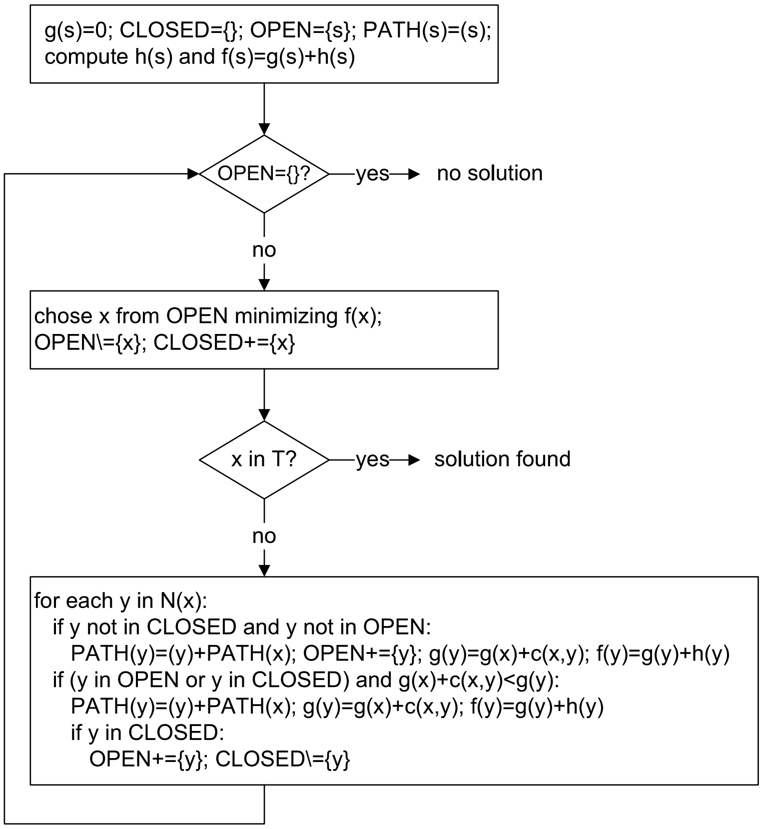
From the figure, we can derive the starting node (1) and the set of target nodes ($T = \{7\}$). Furthermore, we can derive the set of neighbors $N(x)$ for each node x as well as the cost $c(x, y)$ for changing from node x to node y being the arc labels.

Perform an A* search using the following table providing the heuristic estimate $h(x)$:

x	1	2	3	4	5	6	7	8	9
$h(x)$	2	2	9	4	8	8	3	4	8

How do OPEN and CLOSED sets as well as PATH(7) look like when hitting the target?

For this task, use the A* convention introduced in the class as depicted in the following diagram:



3.2 Solution

ID	step	x	$N(x)$	PATH(x)	OPEN	CLOSED	$g(x)$	$f(x)$
1	init	1	{3,4}	(1)	{1}	{}	0	2
2	pick min	1	{3,4}	(1)	{}	{1}	0	2
3	iterate	3	{5,8}	(3,1)	{3}	{1}	9	18
4		4	{2}	(4,1)	{3,4}	{1}	4	8
5	pick min	4	{2}	(4,1)	{3}	{1,4}	4	8
6	iterate	2	{1,6}	(2,4,1)	{2,3}	{1,4}	5	7
7	pick min	2	{1,6}	(2,4,1)	{3}	{1,2,4}	5	7
8	iterate	1				{1,2,4}	7	
9		6	{7}	(6,2,4,1)	{3,6}	{1,2,4}	10	18
10	pick min	3	{5,8}	(3,1)	{6}	{1,2,3,4}	9	18
11	iterate	5	{7}	(5,3,1)	{5,6}	{1,2,3,4}	18	26
12		8	{9}	(8,3,1)	{5,6,8}	{1,2,3,4}	10	14
13	pick min	8	{9}	(8,3,1)	{5,6}	{1,2,3,4,8}	10	14
14	iterate	9	{7}	(9,8,3,1)	{5,6,9}	{1,2,3,4,8}	13	21
15	pick min	6	{7}	(6,2,4,1)	{5,9}	{1,2,3,4,6,8}	10	18
16	iterate	7	{}	(7,6,2,4,1)	{5,7,9}	{1,2,3,4,6,8}	15	18
17	pick min	7	{}	(7,6,2,4,1)	{5,9}	{1,2,3,4,6,7,8}	15	18

Alternative when picking node 6 at ID 10:

ID	step	x	$N(x)$	PATH(x)	OPEN	CLOSED	$g(x)$	$f(x)$
1	init	1	{3,4}	(1)	{1}	{}	0	2
2	pick min	1	{3,4}	(1)	{}	{1}	0	2
3	iterate	3	{5,8}	(3,1)	{3}	{1}	9	18
4		4	{2}	(4,1)	{3,4}	{1}	4	8
5	pick min	4	{2}	(4,1)	{3}	{1,4}	4	8
6	iterate	2	{1,6}	(2,4,1)	{2,3}	{1,4}	5	7
7	pick min	2	{1,6}	(2,4,1)	{3}	{1,2,4}	5	7
8	iterate	1				{1,2,4}	7	
9		6	{7}	(6,2,4,1)	{3,6}	{1,2,4}	10	18
10	pick min	6	{7}	(6,2,4,1)	{3}	{1,2,4,6}	10	18
11	iterate	7	{}	(7,6,2,4,1)	{3,7}	{1,2,4,6}	15	18
12	pick min	7	{}	(7,6,2,4,1)	{3}	{1,2,4,6,7}	15	18

Alternative when picking node 3 at ID 12:

ID	step	x	$N(x)$	PATH(x)	OPEN	CLOSED	$g(x)$	$f(x)$
1	init	1	{3,4}	(1)	{1}	{}	0	2
2	pick min	1	{3,4}	(1)	{}	{1}	0	2
3	iterate	3	{5,8}	(3,1)	{3}	{1}	9	18
4		4	{2}	(4,1)	{3,4}	{1}	4	8
5	pick min	4	{2}	(4,1)	{3}	{1,4}	4	8
6	iterate	2	{1,6}	(2,4,1)	{2,3}	{1,4}	5	7
7	pick min	2	{1,6}	(2,4,1)	{3}	{1,2,4}	5	7
8	iterate	1				{1,2,4}	7	
9		6	{7}	(6,2,4,1)	{3,6}	{1,2,4}	10	18
10	pick min	6	{7}	(6,2,4,1)	{3}	{1,2,4,6}	10	18
11	iterate	7	{}	(7,6,2,4,1)	{3,7}	{1,2,4,6}	15	18
12	pick min	3	{5,8}	(3,1)	{7}	{1,2,3,4,6}	9	18
13	iterate	5	{7}	(5,3,1)	{5,7}	{1,2,3,4,6}	18	26
14		8	{9}	(8,3,1)	{5,7,8}	{1,2,3,4,6}	10	14
15	pick min	8	{9}	(8,3,1)	{5,7}	{1,2,3,4,6,8}	10	14
16	iterate	9	{7}	(9,8,3,1)	{5,7,9}	{1,2,3,4,6,8}	13	21
17	pick min	7	{}	(7,6,2,4,1)	{5,9}	{1,2,3,4,6,7,8}	15	18

Solution with an alternative heuristic estimate $h(x)$:

x	1	2	3	4	5	6	7	8	9
$h(x)$	6	5	2	3	6	0	0	8	3

ID	step	x	$N(x)$	PATH(x)	OPEN	CLOSED	$g(x)$	$f(x)$
1	init	1	{3,4}	(1)	{1}	{}	0	6
2	pick min	1	{3,4}	(1)	{}	{1}	0	6
3	iterate	3	{5,8}	(3,1)	{3}	{1}	9	11
4		4	{2}	(4,1)	{3,4}	{1}	4	7
5	pick min	4	{2}	(4,1)	{3}	{1,4}	4	7
6	iterate	2	{1,6}	(2,4,1)	{2,3}	{1,4}	5	10
7	pick min	2	{1,6}	(2,4,1)	{3}	{1,2,4}	5	10
8	iterate	1				{1,2,4}	7	
9		6	{7}	(6,2,4,1)	{3,6}	{1,2,4}	10	10
10	pick min	6	{7}	(6,2,4,1)	{3}	{1,2,4,6}	10	10
11	iterate	7	{}	(7,6,2,4,1)	{3,7}	{1,2,4,6}	15	15
12	pick min	3	{5,8}	(3,1)	{7}	{1,2,3,4,6}	9	11
13	iterate	5	{7}	(5,3,1)	{5,7}	{1,2,3,4,6}	18	24
14		8	{9}	(8,3,1)	{5,7,8}	{1,2,3,4,6}	10	18
15	pick min	7	{}	(7,6,2,4,1)	{5,8}	{1,2,3,4,6,7}	15	15

4 Expert and dialog systems (5 pts)

4.1 Task

A call router for a banking system contains a module that can ask callers two questions:

q_1) Are you calling about your credit card (C) or your account (A)?

q_2) Do you want to order (O), cancel (L), or change (H) your credit card/account?

The following business logic table of the module shows which combinations of responses to q_1 and q_2 route to which destination d_i ($r_{1,i}$, $r_{2,i}$). Based on statistics drawn from the bank's call centers, for each response combination, historic counts c_i are available.

i	$r_{1,i}$	$r_{2,i}$	d_i	$c_i/1000$
1	C	O	*orderCard*	11
2	C	L	*agent*	17
3	C	H	*changeCard*	36
4	A	O	*agent*	9
5	A	L	*agent*	13
6	A	H	*changeAccount*	14

Considering that a call router should minimize the expected number of user turns and, therefore, should only ask necessary questions, which of the following statements is true:

a) The optimal sequence of questions to ask is q_1, q_2.

b) The optimal sequence of questions to ask is q_2, q_1.

c) The sequence of questions does not make a difference.

Justify your answer and calculate the expected number of user turns (average number of asked questions) for the scenarios a) and b).

4.2 Solution

In the business logic table, we can see that no matter which response is given to q_1 (C or A), we do not know the destination yet, so, q_2 needs to be asked as well. This makes the expected number of user turns

$$E(q_1, q_2) = 2.$$

On the other hand, when q_2 is asked first and the response turns out to be L, we know that the destination is *agent* and we do not need to ask a second question making b) the true statement. In the other cases (O or H) the destination is still ambiguous and q_1 has to be asked as well. Thus, with the probability $p(L)$

we ask one, with $1-p(L)$ two questions which translates to the expected number of user turns for the opposite direction:

$$\begin{aligned} E(q_2, q_1) &= p(L) + 2 \cdot (1 - p(L)) \\ &= 2 - p(L) \\ &= 2 - \frac{17 + 13}{11 + 17 + 36 + 9 + 13 + 14} \\ &= 2 - \frac{30}{100} \\ &= 1.7 \end{aligned}$$