CONCEPTUALIZATION DEFINITION  Conceptualization is step one of formalization of knowledge in declarative form $C = (U, F, P, R)$, where $U$ is a non empty finite set of objects called universe of discourse, $F$ a finite set of functions defined on $U$, $R$ is a finite set of relations defined on $U$.

QUESTION 1  (5pts)

Conceptualize the following situation

In a room there are 3 girls, 2 boys, and 2 cars one red and one blue.

The following properties must be true.  
1. Each girl likes exactly one boy.  
2. Some boys like some girls.  
3. Two boys like a red car.  
4. One girl likes a blue car.

Use as the the universe a set $U = \{ o_1, o_2, o_3, o_4, o_5, o_6, o_7 \}$ and the set relations: $R = \{ \text{GIRL, BOY, CAR, RCAR, BCAR, LIKE} \}$. Use the intended interpretation

Follow the steps below to write your solution.

0. (1pt) Define relations GIRL, BOY, CAR, RCAR, BCAR describing the property In a room there are 3 girls, 2 boys, and 2 cars one red and one blue.

Solution

These are MY definitions - you can have different sets of elements defining the relations.

GIRL = \{o_1, o_2, o_3\}, BOY = \{o_4, o_5\}, CAR= \{o_6, o_7\}, RCAR= \{o_6\}, BCAR= \{o_7\}

Observe that RCAR $\subseteq$ CAR and BCAR $\subseteq$ CAR

1. (1pt) Define relation LIKE 1 that makes property Each girl likes exactly one boy TRUE

Solution

LIKE 1 = \{(o_1, o_4), (o_2, o_4), (o_3, o_5)\}

2. (1pt) Define relation LIKE 2 that makes property Some boys like some girls TRUE

Solution

LIKE 2 = \{(o_4, o_1)\}

3. (1pt) Define relation LIKE 3 that makes property Two boys like a red car TRUE

Solution

LIKE 3 = \{(o_4, o_6), (o_5, o_6)\}

4. (1pt) Define relation LIKE 4 that makes property One girl likes a blue car TRUE

Solution

LIKE 4 = \{(o_2, o_7)\}

Write LIKE = LIKE1 $\cup$ LIKE2 $\cup$ LIKE3 $\cup$ LIKE 4
Solution
LIKE = {(o1, o4), (o2, o4), (o3, o5), (o4, o1), (o4, o6), (o5, o6), (o2, o7)}

RULE BASED SYSTEMS

QUESTION 2 (5pts)
Here is a small set of RULES proposed for a simple rule-based system S for dealing with cars

R1 IF car is broken AND old AND income is small THEN repair it
R2 IF car is broken AND old AND do not repair AND income is medium THEN buy a used car
R3 IF car is broken AND is not old AND income is large THEN buy a new car

P1. (1pts) WRITE the rules R1, R2, R3 of the system S in propositional convention 1, i.e. as rules

\[ A_1 \cap A_2 \cap \cdots \cap A_n \Rightarrow C \]

where \( A_1, A_2, \ldots, A_n, C \) are atomic or negations of atomic formulas

Follow the steps below to write your solution.

1. (0.5 pt) Specify your choice of atomic formulas and negations of atomic formulas needed to represent the rules

Solution

ATOMIC FORMULAS are: A, B, C, D, E, F, G, H

A - represents ” car is broken ”
B - represents ” car is old ”
C - represents ” income is small”
D - represents ” repair car”
E - represents ” income is medium ”
F - represents ”buy a used car ”
G - represents ”income is large ”
H - represents ”buy a new car ”

Negations of ATOMIC FORMULAS are

\( \neg B \) - represents ” car is not old”
\( \neg D \) - represents ” do not repair”

2. (0.5 pt) Specify the rules

Solution

RULES are

R1 \[ A \cap B \cap C \Rightarrow D \]
R2 \[ A \cap B \cap \neg D \cap E \Rightarrow F \]
R3 \[ A \cap \neg B \cap G \Rightarrow H \]

P2. (1pt) WRITE the rules R1, R2, R3 in propositional convention 2 as rules
\[ A_1 \cap A_2 \cap \cdots \cap A_n \Rightarrow C \] 
where \( A_1, A_2, \ldots, A_n, C \) are atomic formulas

Follow the steps below to write your solution.

1. (0.5 pt) Specify your choice of atomic formulas needed to represent the rules

Solution

ATOMIC FORMULAS are: A, B, C, D, E, F, G, H, J, K

A - represents "car is broken"
B - represents "car is old"
C - represents "income is small"
D - represents "repair car"
E - represents "income is medium"
F - represents "buy a used car"
G - represents "income is large"
H - represents "buy a new car"
J - represents "car is NOT old"
K - represents "do not repair car"

2. (0.5 pt) Specify the rules

Solution

RULES are

R1 \[ A \cap B \cap C \Rightarrow D \]
R2 \[ A \cap B \cap K \cap E \Rightarrow F \]
R3 \[ A \cap J \cap G \Rightarrow H \]

P3. (3pts) Here is an expert system S with rules:

R1 IF car is broken AND old AND income is small THEN repair it
R2 IF car is broken AND old AND do not repair AND income is medium THEN buy a used car
R3 IF car is broken AND is not old AND income is large THEN buy a new car.

Follow the steps below to write the rules of S in the predicate convention: attribute(\( x \), attribute value).

1. (1pt) DEFINE all needed ATTRIBUTES and their values.
USE the intended interpretation NAMES for the ATTRIBUTES

**Solution**

I use the intended interpretation names for ATTRIBUTES - you can use your own names

The ATTRIBUTES and their VALUES are:

- **CarBroken** with values yes, no
- **CarOld** with values yes, no
- **Income** with values small, medium, large
- **CarRepair** with values yes, no
- **CarBuy** with values new, used

2. (1pt) WRITE the RULES

**Solution**

RULES ARE:

R1  \( CarBroken(x, yes) \cap CarOld(x, yes) \cap Income(x, small) \Rightarrow CarRepair(x, yes) \)

R2  \( CarBroken(x, yes) \cap CarOld(x, yes) \cap CarRepair(x, no) \cap Income(x, medium) \Rightarrow CarBuy(x, used) \)

R3  \( CarBroken(x, yes) \cap CarOld(x, no) \cap Income(x, large) \Rightarrow CarBuy(x, new) \)

3. (1pt) WRITE a database TABLE with your own example of any 4 records describing some facts in S

**Solution**

There are my records, yours can be different!

<table>
<thead>
<tr>
<th>Obj</th>
<th>CarBroken</th>
<th>CarOld</th>
<th>Income</th>
<th>CarRepair</th>
<th>CarBuy</th>
</tr>
</thead>
<tbody>
<tr>
<td>0₁</td>
<td>yes</td>
<td>yes</td>
<td>small</td>
<td>no</td>
<td>new</td>
</tr>
<tr>
<td>0₂</td>
<td>no</td>
<td>no</td>
<td>large</td>
<td>yes</td>
<td>used</td>
</tr>
<tr>
<td>0₃</td>
<td>yes</td>
<td>yes</td>
<td>medium</td>
<td>no</td>
<td>new</td>
</tr>
<tr>
<td>0₄</td>
<td>yes</td>
<td>no</td>
<td>small</td>
<td>no</td>
<td>used</td>
</tr>
</tbody>
</table>

**RESOLUTION PART**

**QUESTION 3 (5pts)**

**P1.** (1pt) Given a set of clauses

\[ \Delta = \{ \{ a, \neg b \}, \{ a, b, c \}, \{ \neg a, c \}, \{ \neg c, \neg b \} \} \]

Write all possible complementary pairs and all their possible resolvents

**Solution** RESOLUTION HOMEWORK

**P2.** (1pt) Use Resolution Deduction to decide whether the set \( \Delta \) of clauses is unsatisfiable or satisfiable.

\[ \Delta = \{ \{ \neg a, b \}, \{ \neg b \}, \{ a, b \} \} \]
Solution

Consider Resolution Deduction as follows

1. $\Delta = \{\{\neg a, \, b\}, \, \{\neg b\}, \, \{a, \, b\}\}$

2. $\{b\}$  Resolution application on $\{\neg a, \, b\}, \, \{a, \, b\}$

3. $\{\}$  Resolution application on $\{b\}, \, \{-b\}$

$\Delta$ is UNSATISFIABLE

P3. (3pts) Use the **Tautology Deletion Strategy** to decide whether the set $C$ is unsatisfiable or satisfiable.

\[
C = \{\{\neg a, \, a, \, b, \, \neg c\}, \, \{a, \, \neg b, \, c, \, b\}, \, \{\neg a, \, b, \, c\}\}
\]

**Tautology Clause** is any clause containing a PAIR of complementary literals. i.e containing any variable $a$ and its negation $\neg a$. Let $C'$ be obtained from $C$ by removing all **tautology clauses** from $C$.

**Tautology Deletion Strategy Theorem** $C$ is unsatisfiable (satisfiable)  iff  $C'$ is unsatisfiable (satisfiable)

**Solution**

Given $C = \{\{\neg a, \, a, \, b, \, \neg c\}, \, \{a, \, \neg b, \, c, \, b\}, \, \{\neg a, \, b, \, c\}\}$.

We remove tautologies $\{\neg a, \, a, \, b, \, \neg c\}$ and $\{a, \, \neg b, \, c, \, b\}$ and we get an **satisfiable** set of clauses

\[
C' = \{\{\neg a, \, b, \, c\}\}
\]

By Tautology Deletion Strategy Theorem the set of clauses $C$ is also **satisfiable**.

**QUESTION 4  Extra 5pts**

Use the Propositional Resolution to prove that

\[
\models (\neg(a \Rightarrow b) \Rightarrow (a \land \neg b))
\]

Write down and explain carefully each steps in the procedure.

**Solution** in Lecture 6 RESOLUTION page 58