Question 1 [8 marks]

Suppose the clauses for atom \( r \) are

\[
\begin{align*}
  r & : - s, t. \\
  r & : - u. 
\end{align*}
\]

(a) [4 marks] Draw the box model for \( r \). You do not need to include the port names, but you need to include the names for the atoms that the boxes represent.

(b) [2 marks] According to the box model, what happens immediately before atom \( t \) is called?

(c) [2 marks] In Prolog's trace, what is printed immediately after “Redo \( r \)”? (You only need to give as much as can be inferred from the box model.)

Question 2 [8 marks]

Consider the following (partial) derivation of the query \(?w\). Note that the knowledge base is not specified. Fill in the underlined missing answers.

<table>
<thead>
<tr>
<th>Answer clause</th>
<th>Clause resolved</th>
</tr>
</thead>
<tbody>
<tr>
<td>yes : - w</td>
<td>Query</td>
</tr>
<tr>
<td>yes : - s, t</td>
<td>w : - s, t</td>
</tr>
<tr>
<td>yes : - r, t</td>
<td>s : - r</td>
</tr>
<tr>
<td>yes : - q, z, t</td>
<td>(a) ____________</td>
</tr>
<tr>
<td>yes : - z, t</td>
<td>(b) ____________</td>
</tr>
<tr>
<td>(c) ____________</td>
<td>z : - u, p</td>
</tr>
<tr>
<td>yes : - n, o, p, t</td>
<td>u : - n, o</td>
</tr>
</tbody>
</table>

(d) If the proof then fails, what does this tell us about the knowledge base?

Question 3 [6 marks]

Consider the following knowledge base. (Assume there are dynamic declarations so there are no errors).

\[
\begin{align*}
  \text{flies_autonomously} & : - \text{bird}, \ \bot \ \text{abfly}. \\
  \text{flies} & : - \text{flies_autonomously}, \ \bot \ \text{injured}.
\end{align*}
\]
flies :- on_plane, \+ plane_broken.
abfly :- emu.
abfly :- penguin.
bird :- emu.
bird :- penguin.
emu.
on_plane.

Give the set of all atoms and negations of atoms that are a logical consequence (i.e., the atoms and their negations that would be produced by the bottom-up proof procedure for negation as failure). You do not need to give the derivation.

**Question 4 [10 marks]**

Suppose that dates are represented as \(ce(Y, M, D)\) for dates in the common era, where they year is \(Y\), the month is \(M\) and the day in the month is \(D\); these are all integers. For example today’s date is \(ce(2017, 9, 27)\).

Write a predicate `next_day(C1, C2)` that is true when date \(C2\) is the date after \(C1\). You can assume the following predicates are predefined:

- \(<\) which compares two arithmetic expressions for “less than”
- \(\text{is}\), where \(V\) is \(E\) is true if arithmetic expression \(E\) evaluates to number \(V\)
- \(\text{number_days}(M, D)\) which is true if month \(M\) contains \(D\) days defined by
  
  \[
  \text{number_days}(M, 31) :- \text{member}(M, [1, 3, 5, 7, 8, 10, 12]) .
  \text{number_days}(2, 28) .
  \text{number_days}(M, 30) :- \text{member}(M, [4, 6, 9, 11]).
  \]

You can assume that \(C1\) does not contain variables when called.

An example of its use is:

```
?- next_day(ce(2017,12,30),D).
D = ce(2017, 12, 31) .
```

```
?- next_day(ce(2017,12,30),D), next_day(D,D1).
D = ce(2017, 12, 31),
D1 = ce(2018, 1, 1) .
```

**Question 5 [10 marks]**

(a) [6 marks] Write a program `replace(Old, New, Lst, Result)` which is true when \(Result\) is a list with the same elements as list \(Lst\) (in the same order) but with all instances of \(Old\) replaced by \(New\). For example, it should have the following behaviour:

```
?- replace(a, w, [a, v, a, t, a, r], R).
R = [w, v, a, t, w, r]
?- replace(w, a, [a, v, a, t, a, r], R).
R = [a, v, a, t, a, r]
```
You may use the predicate \( \text{dif}(X, Y) \) which is true when \( X \) is different to \( Y \), but no other built-in predicates.

(b) [4 marks] What are all of the answers to the query:

\[
?- \text{replace(prolog,fun,L,[fun,is,fun])}.
\]

**Question 6 [3 marks]**

(a) What do you like about the course so far?
(b) What do you dislike about the course so far?
(c) What should be changed about this course?