Exercise 1 (Simple Prolog): (1.5 + 2 + 1.5 = 5 points)

Consider the following Prolog program, which realizes simple terms built from variables and the function symbol plus.

\begin{verbatim}
variable(x).
variable(y).
term(X) :- variable(X).
term(plus(X,Y)) :- term(X), term(Y).
\end{verbatim}

a) Implement a predicate \texttt{uses(TERM, A)} in Prolog which is true iff \texttt{TERM} is a term built with \texttt{plus} and \texttt{A} is an argument of the outermost \texttt{plus} symbol of \texttt{TERM}, i.e., \texttt{term(TERM)} is true and moreover, \texttt{TERM = plus(A,\_)} or \texttt{TERM = plus(\_,A)}.

b) Implement a predicate \texttt{contains(TERM, X)} in Prolog which is true iff \texttt{TERM} is a term built with \texttt{plus} and \texttt{X} is a subterm of \texttt{TERM}. In other words, \texttt{contains(TERM, X)} is true iff \texttt{uses(TERM, X)} is true or if there are \( n > 0 \) elements \( Y_1, \ldots, Y_n \) such that the following statements are true:

\begin{itemize}
  \item \texttt{uses(TERM,Y_1)}
  \item \texttt{uses(Y_n, X)}
  \item \texttt{uses(Y_{i-1},Y_i)} for all \( i \in \{2, \ldots, n\} \).
\end{itemize}

Make sure that the evaluation of all queries \texttt{?- contains(t1,t2)} terminates if \texttt{t1} and \texttt{t2} are ground terms.

c) List the first five answers that Prolog computes for the following queries, in the order that Prolog gives them. Try to solve this part of the exercise without the help of a computer.

1. \texttt{?- term(plus(x,X)).}
2. \texttt{?- uses(plus(x,y), X).}
3. \texttt{?- contains(plus(plus(x,y),x), X).}

Solution:

\begin{verbatim}
a) uses(T,A) :- term(T), uses_plus(T,A).
   uses_plus(plus(A,\_),A).
   uses_plus(plus(\_,A),A).
b) contains(T,A) :- uses(T,A).
   contains(T,A) :- uses(T,TP), contains(TP,A).
c) 1. X = x ;
    X = y ;
    X = plus(x,x) ;
    X = plus(x,y) ;
    X = plus(x,plus(x,x)) ;

  2. X = x ;
     X = y.

  3. X = plus(x,y) ;
     X = x ;
     X = x ;
     X = y ;
     false.
\end{verbatim}
Exercise 2 (Syntax): 

Consider the following Prolog program.

```
robot(wall_e).
robot(otto).
robot(c3po).
robot(r2d2).
robot(android(looks_like_a_human)).
robot(android(looks_like_a_machine)).
can_walk(c3po).
can_drive(r2d2).
can_drive(wall_e).
same_story(c3po,r2d2).
same_story(wall_e,otto).

can_move(X) :- can_walk(X).
can_move(X) :- can_drive(X).
same_story(X,Y) :- robot(X), robot(Y), same_story(Y,X).
```

a) Construct the corresponding sets of formulas, predicate symbols, function symbols, and variables based on the program.

b) Give Prolog queries corresponding to the following questions:
   
   • “Which robots can both walk and drive?”
   • “Which pairs of robots can move, and are both part of the same story?”

Solution:

a) $\Phi = \{\$

```
robot(wall_e),
robot(otto),
robot(c3po),
robot(r2d2),
robot(android(looks_like_a_human)),
robot(android(looks_like_a_machine)),
can_walk(c3po),
can_drive(r2d2),
can_drive(wall_e),
same_story(c3po,r2d2),
same_story(wall_e,otto),
\forall X \text{ can\_walk}(X) \rightarrow \text{can\_move}(X),
\forall X \text{ can\_drive}(X) \rightarrow \text{can\_move}(X),
\forall X, Y \text{ robot}(X) \land \text{robot}(Y) \land \text{same\_story}(Y,X) \rightarrow \text{same\_story}(X,Y)
```

\}$ over $\Sigma_0 = \{\text{wall\_e, otto, c3po, r2d2, looks\_like\_a\_human, looks\_like\_a\_machine}\}$, $\Sigma_1 = \{\text{android}\}$, $\Sigma = \Sigma_0 \cup \Sigma_1$, $\Delta_2 = \{\text{same\_story}\}$, $\Delta_1 = \{\text{robot, can\_walk, can\_drive, can\_move}\}$, $\Delta = \Delta_1 \cup \Delta_2$, and $V = \{X, Y\}$.

b) $\bullet$ ?- robot(X), can_walk(X), can_drive(X).
• ?- robot(X), robot(Y), same_story(X, Y), can_move(X), can_move(Y).

Exercise 3 (Induction): (4 points)

Let \( t \) be an arbitrary term. Then the depth \( d(t) \) of \( t \) is defined as follows: \( d(X) = 1 \) if \( X \) is a variable. Otherwise we have for \( n \geq 0 \) that \( d(f(t_1, \ldots, t_n)) = 1 + \max\{d(t_1), \ldots, d(t_n)\} \).

Show by structural induction that for every term \( t \) and every variable renaming \( \sigma \) we have \( d(t) = d(\sigma(t)) \).

Solution:

Let \( t \) be an arbitrary term and \( \sigma \) be an arbitrary variable renaming. In the induction base, we consider the case that \( t \) is a variable. Then we have \( d(t) = 1 \). According to the definition of variable renamings, we have \( \sigma(t) \in V \) with \( d(\sigma(t)) = 1 \), so \( d(t) = 1 = d(\sigma(t)) \) holds.

In the induction step, we consider the case \( t = f(t_1, \ldots, t_n) \) with \( n \geq 0 \). As induction hypothesis, we can assume that we have \( d(t_i) = d(\sigma(t_i)) \) for all \( i \in \{1, \ldots, n\} \). We have \( \sigma(t) = f(\sigma(t_1), \ldots, \sigma(t_n)) \). We also know that \( d(t) = 1 + \max\{d(t_1), \ldots, d(t_n)\} \). Using the induction hypothesis, we then have \( d(t) = 1 + \max\{d(t_1), \ldots, d(t_n)\} = 1 + \max\{d(\sigma(t_1)), \ldots, d(\sigma(t_n))\} = d(\sigma(t)) \).

Exercise 4 (Semantics): (3 + 3 + 3 = 9 points)

Let \( (\Sigma, \Delta) \) be a signature with \( \Sigma = \Sigma_0 \cup \Delta_1 \cup \Delta_3, \Delta_1 = \{\text{even}\}, \Delta_3 = \{\text{plus}\} \) and \( \Sigma_0 = \{0, 1, 2\} \).

Moreover, let

- \( \Phi = \{\text{even}(0), \text{even}(2), \forall X, Y, Z \text{ even}(Z) \land \text{plus}(X, Y, Z) \rightarrow \text{even}(X) \land \text{even}(Y)\} \),
- \( \varphi = \forall Z \text{ plus}(2, 1, Z) \land \text{even}(Z) \rightarrow \text{even}(1) \),
- \( S = (\mathbb{N}, \alpha) \) with 
  - \( \alpha_0 = 0, \alpha_1 = 1, \alpha_2 = 2 \),
  - \( \alpha_{\text{plus}} = \{ (x, y, z) \in \mathbb{N}^3 \mid x + y = z \} \),
  - \( \alpha_{\text{even}} = \{ 2 \ast i \mid i \in \mathbb{N} \} \).

Prove or disprove the following statements.

You may use that addition on natural numbers is commutative.

\begin{align*}
\text{a) } S & \models \varphi \\
\text{b) } & \models \varphi \\
\text{c) } \Phi & \models \varphi 
\end{align*}

Solution: __________


a) Let $I = (\mathbb{N}, \alpha, \beta)$ for some variable assignment $\beta$. Since $\varphi$ contains no free variables, we have:

\[
S \models \varphi \iff I \models \varphi
\]

\[
I \models \forall Z \ (\text{plus}(2, 1, Z) \land \text{even}(Z) \rightarrow \text{even}(1))
\]

\[
\left(I[Z/z] \models \text{plus}(2, 1, Z) \land \text{even}(Z) \rightarrow \text{even}(1)\right) \quad \text{for all } z \in \mathbb{N}
\]

\[
\left(I[Z/z] \models \text{even}(1) \text{ if } I[Z/z] \models \text{plus}(2, 1, Z) \land \text{even}(Z)\right) \quad \text{for all } z \in \mathbb{N}
\]

\[
\left(I[Z/z] \models \text{even}(1) \text{ if } I[Z/z] \models \text{plus}(2, 1, Z) \text{ and } I[Z/z] \models \text{even}(Z)\right) \quad \text{for all } z \in \mathbb{N}
\]

\[
\left(\alpha_1 \in \alpha_{\text{even}} \text{ if } ((\alpha_2, \alpha_1, z) \in \alpha_{\text{plus}} \land z \in \alpha_{\text{even}})\right) \quad \text{for all } z \in \mathbb{N}
\]

\[
\left(1 \in \{2 \ast i \mid i \in \mathbb{N}\} \text{ if } \right.
\]

\[
\left((2, 1, z) \in ((x, y, z) \in \mathbb{N}^3 \mid x + y = z) \text{ and } z \in \{2 \ast i \mid i \in \mathbb{N}\}\right) \quad \text{for all } z \in \mathbb{N}
\]

\[
\left(1 \in \{2 \ast i \mid i \in \mathbb{N}\} \text{ if } (3 = z \text{ and } z \in \{2 \ast i \mid i \in \mathbb{N}\}) \right) \quad \text{for all } z \in \mathbb{N}
\]

\[
1 \in \{2 \ast i \mid i \in \mathbb{N}\} \text{ if } 3 \in \{2 \ast i \mid i \in \mathbb{N}\}
\]

This implication is obviously true since both 1 and 3 are not divisible by two.

b) We choose the following interpretation $I' = (\{0, 1, 2\}, \alpha', \beta)$ with some variable assignment $\beta$ and $\alpha'_0 = 0, \alpha'_1 = 1, \alpha'_2 = 2, \alpha'_{\text{plus}} = \{(2, 1, 0)\}, \alpha'_{\text{even}} = \{0\}$. Then we have:

\[
I' \models \varphi \iff I' \models \forall Z \ (\text{plus}(2, 1, Z) \land \text{even}(Z) \rightarrow \text{even}(1))
\]

\[
\left(I'[Z/z] \models \text{plus}(2, 1, Z) \land \text{even}(Z) \rightarrow \text{even}(1)\right) \quad \text{for all } z \in \{0, 1, 2\}
\]

\[
I'[Z/0] \models \text{plus}(2, 1, Z) \land \text{even}(Z) \rightarrow \text{even}(1)
\]

\[
\left(I'[Z/0] \models \text{plus}(2, 1, Z) \text{ and } I'[Z/0] \models \text{even}(Z), \text{ then } I'[Z/0] \models \text{even}(1)\right)
\]

\[
\left(I'[Z/0] \models \text{even}(1) \text{ if } (\alpha'_2, \alpha'_1, 0) \in \alpha'_{\text{plus}} \text{ and } 0 \in \alpha'_{\text{even}}, \text{ then } \alpha'_1 \in \alpha'_{\text{even}}\right)
\]

\[
\left(I'[Z/0] \models \text{even}(1) \text{ if } (2, 1, 0) \in \{(2, 1, 0)\} \text{ and } 0 \in \{0\}, \text{ then } 1 \in \{0\}\right)
\]

This is obviously wrong. Thus, the statement is disproved as $I'$ is not a model of $\varphi$.

c) Let $I'' = (U, \alpha'', \beta'')$ be an arbitrary interpretation for the given signature. If we can show that $I'' \models \Phi$, implies $I'' \models \varphi$, then we have $\Phi \models \varphi$.

If $I'' \models \varphi$, the implication trivially holds. Therefore, assume that $I'' \models \Phi$. We observe that:

\[
I'' \models \Phi \iff I'' \models \text{even}(0) \land \text{even}(2) \land \forall X, Y, Z \ (\text{even}(Z) \land \text{plus}(X, Y, Z) \rightarrow \text{even}(X) \land \text{even}(Y))
\]

\[
I'' \models \text{even}(0) \text{ and } I'' \models \text{even}(2) \text{ and }
\]

\[
I'' \models \forall X, Y, Z \ (\text{even}(Z) \land \text{plus}(X, Y, Z) \rightarrow \text{even}(X) \land \text{even}(Y))
\]

\[
I'' \models \forall X, Y, Z \ (\text{even}(Z) \land \text{plus}(X, Y, Z) \rightarrow \text{even}(X) \land \text{even}(Y))
\]

\[
\left(I''[X/x, Y/y] \models \forall Z \ (\text{even}(Z) \land \text{plus}(X, Y, Z) \rightarrow \text{even}(X) \land \text{even}(Y)) \right) \quad \text{for all } x, y \in U
\]

\[
I''[X/\alpha'_1, Y/\alpha'_2] \models \forall Z \ (\text{even}(Z) \land \text{plus}(X, Y, Z) \rightarrow \text{even}(X) \land \text{even}(Y))
\]

\[
I'' \models \forall Z \ (\text{even}(Z) \land \text{plus}(2, 1, Z) \rightarrow \text{even}(1))
\]

\[
I'' \models \varphi
\]

Therefore, the statement is true.