1.) Given an undirected graph $G = (V, E)$, we call a set $S \subseteq V$ self-defending in $G$ if for each $v \in S$ and $u \in V$ with $(u, v) \in E$, there exists a $w \in S$ with $(w, u) \in E$.

Consider the following decision problem:

**SELF-DEFENSE(SD)**

INSTANCE: A directed graph $G = (V, E)$ and two vertices $a, b \in V$.

QUESTION: Does there exist a set self-defending set $S \subseteq V$ in $G$ with $a \in S$ and $b \notin S$.

(a) The following function $f$ provides a polynomial-time many-one reduction from SD to SAT: for an instance $I = ((V, E), a, b)$ of SD let $f(I) = \varphi$ over atoms $x_v$ ($v \in V$):

$$
\varphi = x_a \land \neg x_b \land \bigwedge_{v \in V} \left( \neg x_v \lor \bigwedge_{(u,v) \in E} \bigvee_{(w,u) \in E} x_w \right)
$$

It holds that $I$ is a yes-instance of SD $\iff$ $f(I)$ is a yes-instance of SAT.

Show the $\implies$ direction of the statement.

(9 points)

(b) In what follows assume the reduction from SD to SAT is correct, and recall that SAT is NP-complete.

Tick the correct statements (for ticking a correct statement a certain number of points is given; ticking an incorrect statement results in a substraction of the same amount; you cannot go below 0 points):

- SD is NP-complete
- SD is NP-hard
- SD is in NP
- SD is in NP, but not in P
- for any NP-complete problem, there exists a polynomial-time many-one reduction from SD to that problem
- for any NP-complete problem, there exists a polynomial-time many-one reduction from that problem to SD

(6 points)
2.) We consider the function $P$ which was introduced by Rózsa Péter in 1935.

**Algorithm 1:** The function $P$

Input: $x, y$, two non-negative integers

Output: The computed integer value for $x, y$

1. if $x == 0$ then
   2. return $2y + 1$;
2. else if $y == 0$ then
   4. return $P(x - 1, 1)$;
3. else return $P(x - 1, P(x, y - 1))$;

(a) Let $\mathbb{N}_0$ denote the set of natural numbers including 0. Use well-founded induction to show

$$\forall x \forall y \ (x \in \mathbb{N}_0 \land y \in \mathbb{N}_0) \rightarrow P(x, y) > x + y).$$

(12 points)

(b) Suppose $P_C$ is an implementation of $P$ in the C programming language with $x$ and $y$ of type unsigned integers of size 32 bit (i.e., of type `uint32_t`). Is

$$P(x', y') = P_C(x', y')$$

true for all integers $x', y'$ satisfying $0 \leq x', y' \leq \text{UINT32_MAX}$, where \text{UINT32_MAX} is the largest value for a variable of type `uint32_t`?

If so, then prove this fact. Otherwise provide a counterexample with an exact explanation of what is computed and what is happening. (3 points)
3.) (a) Let \( p \) be the following IMP program:

\[
\begin{align*}
x &:= 0; y := 0; \\
\text{while } y < n \text{ do} \\
&\quad x := x - 6 \ast y - 3; \\
&\quad y := y + 1 \\
\text{od}
\end{align*}
\]

Give a loop invariant and variant for the \texttt{while} loop in \( p \) and prove the validity of the total correctness triple \([n = 10] \ p \ [x + 300 = 0]\).

(10 points)

(b) Let \( p \) be the following IMP program:

\[
\begin{align*}
\text{if } x < y \text{ do} \\
&\quad a := y; \\
&\quad y := x; \\
&\quad x := a; \\
\text{od} \\
x &:= y - x; \\
z &:= z + x \ast y
\end{align*}
\]

Given the program \( p \) above, is it true, that the triple \( \{A\} \ p \ {B}\) is valid if and only if \( VC(p, B) \land (A \implies wlp(p, B)) \)? Briefly justify your answer.

(2 points)

(c) Let \( p \) be the following IMP program containing an integer-valued program variable \( x \):

\[
\begin{align*}
i &:= 0 \\
\text{while } x > 0 \text{ do} \\
&\quad i := i + 1 \\
\text{od}
\end{align*}
\]

Consider the invalid Hoare triple \( \{true\} \ p \ {false}\). Which of the following counterexamples is correct? Tick all the boxes of program states denoting valid counterexamples to the above triple. [Each correct box counts one point, that is you can lose points for incorrectly ticking or leaving the box empty with a minimum of 0 points. You will not lose points for other exercises.]

\( \square \sigma(x) = 0 \quad \square \sigma(x) = 1 \quad \square \sigma(x) = -1 \)

(3 points)
4.) (a) **Simulation**

Provide a non-empty simulation relation $H$ that witnesses $M_1 \leq M_2$, where $M_1$ and $M_2$ are shown below. The initial state of $M_1$ is $s_0$, the initial state of $M_2$ is $t_0$:

**Kripke structure $M_1$:**

- $s_0: \{a\}$
- $s_1: \{b\}$
- $s_2: \{a\}$
- $s_3: \{c\}$
- $s_4: \{b\}$

**Kripke structure $M_2$:**

- $t_0: \{a\}$
- $t_1: \{b\}$
- $t_2: \{c\}$
- $t_3: \{b\}$
- $t_4: \{c\}$
- $t_5: \{b\}$

(4 points)
(b) **CTL Marking Algorithm**

Consider the following Kripke structure $M$:

![Diagram](#)

Execute the **CTL Marking Algorithm** to determine which states $s_i$ satisfy the formulae $\Phi$.

i. $\text{EXEX}a$, and

ii. $\text{AF}(\neg c)$.

In particular,

i. Transform $\Phi$ into an equivalent formula $\Phi'$ in the *existential fragment* of CTL.

ii. List the subformulae of $\Phi'$.

iii. For increasing nesting depth $i$, iteratively give the states $s_i$ marked by subformulae $\phi_0, \psi_0, \phi_1, \psi_1, \ldots$ of $\Phi'$.

iv. Finally, give the return value of the Marking Algorithm. That is, list the states $s_i$ that satisfy formula $\Phi$, i.e., for which states do we have that $M, s_i \models \Phi$?

**Hint:** Recall that the algorithm starts by marking propositional atoms $\phi_0$. It then iteratively marks boolean combinations $\psi_i$ of subformulas $\phi_i$, and temporal operator applications $\phi_{i+1} = \circ \psi_i$ where $\circ \in \{\text{EF}, \text{EU}, \text{EG}, \text{EX}\}$.

i) Answer template for $\Phi = \text{EXEX}a$

Subformulae of $\Phi$: __________________________________________________________________________

Annotate the states of $M$ with the subformulae by which the Marking Algorithm marks them:

![Diagram](#)

States satisfying $\Phi$: __________________________________________________________________________
ii) Answer template for $\Phi = \text{AF}(\neg c)$

Equivalent existential formula $\Phi' \equiv \Phi$: ______________________________

Subformulae of $\Phi'$: ______________________________

Annotate the states of $M$ with the subformulae by which the Marking Algorithm marks them:

![Diagram]

States satisfying $\Phi$: ______________________________

(7 points)
(c) **LTL tautologies**

Prove that the following formulas are tautologies, i.e., they hold for every Kripke structure $M$ and every path $\pi$ in $M$, or find a Kripke structure $M$ and path $\pi$ in $M$, for which the formula does not hold and justify your answer.

i. $(F a) \land (F b) \Rightarrow F (a \land b)$

ii. $(G a) \lor (G b) \Rightarrow G (a \lor b)$

(4 points)