EPFL Formal Verification Course Exam, 28 November 2024

IMPORTANT INFORMATION

Do not open the exam until we tell you to. The exam is three hours long. Place your CAMIPRO card on your desk. Put all electronic devices in a bag away from bench. Write using permanent, dark pen (no graphite nor heat-disappearing pen unless you were granted an exception). Write answers to different problems on disjoint sheets of paper that we supply. Write your name, SCIPER and question number on the top-right of each sheet you return. Do not write the solutions that you want us to grade on the sheets with exam questions; please take these printed exam sheets with you after the exam. You are only allowed two A4-sized two-sided cheat sheets (four A4 pages in total).

The maximal number of points on the exam is 40. We advise you to first solve questions that you find easier. If you are running out of time on a particular problem, try to convince us that you know the right strategy to solve it.

Reminder: If $t \subseteq B \times B$ is a binary relation and $C \subseteq B$, we define

$$t[C] = \{y \mid \exists x \in C.\ (x, y) \in t\}$$

The diagonal relation on $C \subseteq B$ is $\Delta_C = \{(x, x) \mid x \in C\}$. Given $t_1, t_2 \subseteq B \times B$, relation composition $\circ$ is:

$$t_1 \circ t_2 = \{(x, z) \mid \exists y.(x, y) \in t_1 \land (y, z) \in t_2\}$$

Reflexive transitive closure of t is

$$t^* = \bigcup_{i \geq 0} t^i$$

where $t^0 = \Delta_B$, $t^1 = t$, $t^{n+1} = t \circ t^n$. If $M = (S, I, r, A)$ is a transition system we define

$$\bar{r} = \{(s, s') \mid \exists a \in A.(s, a, s') \in r\}$$

and define the reachable states as image of I under the reflexive transitive closure of $\bar{r}$:

$$Reach(M) = \bar{r}^*[I]$$

Do not open the exam until we tell you to.

1 Lattices (10 points)

Let $(L, \sqsubseteq, \sqcup, \sqcap)$ be a lattice with the least element $\bot$. Suppose that L is such that every increasing chain of elements indexed by natural numbers, $x_0 \sqsubseteq x_1 \sqsubseteq \ldots$ has a least upper bound $\bigsqcup_{i \in \mathbb{N}} x_i$ in L. (You may not assume that the lattice is complete unless you can prove that first.) Let $f : L \to L$ be a function that is monotonic with respect to $\sqsubseteq$. Define the sequence $a_i \in L$, for $i \in \mathbb{N}$ inductively by:

$$\begin{array}{lcl} a_0 & = & \bot \\ a_{i+1} & = & f(a_i) \end{array}$$

Define $g : L \to L$ by $g(x) = x \sqcup f(x)$ and the sequence $b_i \in L$, $i \in \mathbb{N}$ by:

$$\begin{array}{lcl} b_0 & = & \bot \\ b_{i+1} & = & g(b_i) \end{array}$$

Let $\mathsf{Fix}_f, \mathsf{Fix}_g \subseteq L$ be the sets of fixed points of f and g, respectively. Let $\mathsf{Post}_f \subseteq L$ be the set of postfix points of f, that is, $\mathsf{Post}_f = \{x \mid f(x) \sqsubseteq x\}$.

Prove or give a counterexample for the following properties.

1. (1pt) g is monotonic with respect to $\sqsubseteq$.

   Solution:

   True. Observe first that the least upper bound operator $\sqcup$ is monotonic: if $x_1 \sqsubseteq x_2$ then $x_1 \sqcup y \sqsubseteq x_2 \sqcup y$. Indeed, $x_2 \sqcup y$ is an upper bound on y and also on x_1 as $x_1 \sqsubseteq x_2 \sqsubseteq x_2 \sqcup y$. Because $x_1 \sqcup y$ is the least *upper bound on these two elements, we have $x_1 \sqcup y \sqsubseteq x_2 \sqcup y$.*

   By commutativity of $\sqcup$, we also have that, if $x_1 \sqsubseteq x_2$ and $y_1 \sqsubseteq y_2$ then

   $$x_1 \sqcup y_1 \sqsubseteq x_2 \sqcup y_1 \sqsubseteq x_2 \sqcup y_2$$

   To prove that g is monotonic, let $x_1 \sqsubseteq x_2$. Then, by monotonicity of f we have $f(x_1) \sqsubseteq f(x_2)$ and monotonicity of $\sqcup$:

   $$g(x_1) = x_1 \sqcup f(x_1) \sqsubseteq x_2 \sqcup f(x_2) = g(x_2)$$

   ◊

2. (2pt) $\forall i \in \mathbb{N}.\ a_i = b_i$. **Solution:** *True. First, note that, because f is monotonic, we have $a_i \sqsubseteq f(a_i)$ for all i, as shown in the lectures. Also, whenever $x \sqsubseteq y$ we have $x \sqcup y = y$, by the definition of the least upper bound, so $a_i \sqcup f(a_i) = f(a_i)$, for all i. We now prove the desired property by induction. For $i = 0$, $a_0 = \bot = b_0$. For the inductive step, suppose $a_i = b_i$. Then*

$$b_{i+1} = g(b_i) = g(a_i) = a_i \sqcup f(a_i) = f(a_i) = a_{i+1}$$

$\Diamond$

3. (1pt) $\forall x \in L.\ x \sqsubseteq g(x)$. **Solution:** *True. By definition of least upper bound, $x \sqsubseteq x \sqcup f(x) = g(x)$.* $\Diamond$

4. (1pt) $\mathsf{Fix}_g = \mathsf{Post}_f$. **Solution:** *True.*

$$x \in \mathsf{Fix}_g \iff g(x) = x \iff x \sqcup f(x) = x \iff f(x) \sqsubseteq x \iff x \in \mathsf{Post}_f$$

$\Diamond$

5. (2pt) If f is ω-continuous then g is also ω-continuous. **Solution:** *True. Consider $x_1 \sqsubseteq x_2 \sqsubseteq \ldots$. Noting that least upper bounds apply to sets, which are unordered collections of elements, and because f is omega-continuous, we have (where $i \in \mathbb{N}$ in all cases):*

$$g(\bigsqcup_i x_i) = \bigsqcup_i x_i \sqcup f(\bigsqcup_i x_i) = \bigsqcup_i x_i \sqcup \bigsqcup_i f(x_i) = \bigsqcup_i (x_i \sqcup f(x_i) = \bigsqcup_i g(x_i)$$

$\Diamond$

6. (2pt) If f is ω-continuous then both f and g have the least fixpoint and these fixpoints are equal. **Solution:** *True. As f is ω-continuous, the least fixed point of f is $\sqcup_{i \in \mathbb{N}} a_i$. As g is ω-continuous, the least fixed point of g is $\sqcup_{i \in \mathbb{N}} b_i$, which is the same because $a_i = b_i$ as proven in part 2.* $\Diamond$

7. (1pt) For $L = \{\bot, \top\}$ with $\bot \sqsubseteq \top$ we have $\mathsf{Fix}_f = \mathsf{Fix}_g$. **Solution:** *False. Define $f(x) = \bot$. Then $\mathsf{Fix}_f = \{\bot\}$ whereas $\mathsf{Fix}_g = \{\bot, \top\}$.* $\Diamond$

2 Quantifier Elimination (5 points)

Use quantifier elimination to find and simplify a quantifier-free Presburger arithmetic formula equivalent to the following one:

$$\forall y.\ \big(\ 3|(y+1)\ \vee\ 3|(y+2)\ \vee\ y < x\ \vee\ 2x \le y\big)$$

Show key steps. All variables range over integers. You may use any valid equivalence-preserving transformations on formulas interpreted over integers, but you need to state the equivalence rule you use explicitly. You may use parametric conjunctions and disjunctions in your intermediate steps ($\bigwedge_{i=\ldots}$, $\bigvee_{i=\ldots}$) but your final result should be written without such parametric operators, and should be as short as possible.

Solution: *We have the following equivalences:*

$$\begin{array}{l}
\forall y.\ \big(\ 3|(y+1)\ \vee\ 3|(y+2)\ \vee\ y < x\ \vee\ 2x \le y\big)\\
\neg\exists y.\ \big(\neg 3|(y+1)\ \wedge\ \neg 3|(y+2)\ \wedge\ x \le y\ \wedge\ y < 2x\big)\\
\neg\exists y.\ \big(3|y\ \wedge\ x-1 < y\ \wedge\ y < 2x\big)\\
\neg\bigvee\limits_{i=1}^{3}\big(3|x-1+i\ \wedge\ x-1 < x-i+i\ \wedge\ x-1+i < 2x\big)\\
\neg\bigvee\limits_{i=1}^{3}\big(3|x-1+i\ \wedge\ i-1 < x\big)\\
\neg\big((3|x \wedge 0 < x) \vee (3|x+1 \wedge 1 < x) \vee (3|x+2 \wedge 2 < x)\big)
\end{array}$$

The following further equivalences also hold. By applying negation:

$$(\neg 3|x \vee x < 1) \wedge (\neg 3|x+1 \vee x < 2) \wedge (\neg 3|x+2 \vee x < 3)$$

Now observe that if $x < 1$, all three conjuncts are true because then also $x < 2$ and $x < 3$. On the other hand, if $x \ge 3$, then for the formula to hold we would have to satisfy all of $\neg 3|x$, $\neg 3|x+1$, $\neg 3|x+2$, which is impossible as one of the three numbers must be divisible by 3. Consequently, aside from $x < 1$, the only possibilities for x are $x = 1$ and $x = 2$. Substituting $x = 1$ makes all conjuncts true. Substituting $x = 2$ makes the second conjunct false. Therefore, the formula is, in fact, equivalent to

$$x \le 1$$

◊

3 Resolution Proof (5 points)

Let R be a predicate of arity two in first-order logic. We say that a formula is closed it has no free variables.

1. (1pt) Write a closed FOL formula stating that R denotes a transitive relation: if x is in relation to y and y is in relation to z, then also x is in relation to z. Write a FOL clause corresponding to the formula above, denote it by A1.

   Solution: *The formula is*

   $$\forall x.\forall y.\forall z.\,(R(x,y) \land R(y,z)) \to R(x,z)$$

   which is equivalent to $\forall x \forall y \forall z\, \neg R(x,y) \lor \neg R(y,z)) \lor R(x,z)$. *Its clausal form is* $A1 = \{\neg R(x,y), \neg R(y,z), R(x,z)\}$. $\Diamond$

2. (1pt) Write a closed FOL formula stating that R denotes an irreleflexive relation: no element is related to itself. Write a FOL clause corresponding to the formula above, denote it by A2.

   Solution: *The formula is*

   $$\forall x.\, \neg R(x,x)$$

   Its clausal form is $A2 = \{\neg R(x,x)\}$. $\Diamond$

3. (1pt) Write a closed FOL formula stating that it is never the case that, simulaneously, x is in relation to y and y is in relation to x. Denote this formula by G.

   Solution:

   $$G := \forall x.\forall y.\, \neg R(x,y) \lor \neg R(y,x)$$

   $\Diamond$

4. (2pt) Write a resolution proof that uses appropriate clauses and resolution rule steps to establish that the formula $A1 \land A2 \to G$ is valid. Each step should indicate which clauses it uses and which variable substitutions (if any) it applies to them.

   Solution: *First, some intuition about the proof, which can be useful for writing the resolution proof.* $A1$ *states that* R *is transitive, which implies in* *particular that if both $R(x, y)$ and $R(y, x)$ hold, then $R(x, x)$ must hold. But this contradicts irreflexivity, which is $A2$.*

To show that $A1 \wedge A2 \rightarrow G$ holds, we show that its negation $A1 \wedge A2 \wedge \neg G$ is unsatisfiable, i.e., we derive the empty clause from $A1$, $A2$ and $\neg G$. We first put $\neg G$ in clausal form:

$$\begin{aligned} \neg G &\iff \neg \forall x. \forall y. \neg R(x, y) \vee \neg R(y, x) \\ &\iff \exists x. \exists y. R(x, y) \wedge R(y, x) \end{aligned}$$

which we Skolemize to obtain the clauses $G1 := \{R(c_x, c_y)\}$ and $G2 := \{R(c_y), R(c_x)\}$ for some constants c_x and c_y. Our resolution proof is thus (with step numbering that starts from -3):

$-3.$ *(A1):* $\{\neg R(x, y), \neg R(y, z), R(x, z)\}$

$-2.$ *(A2):* $\{\neg R(x, x)\}$

$-1.$ *(G1):* $\{R(c_x, c_y)\}$

$0.$ *(G2):* $\{R(c_y), R(c_x)\}$

1. $\{\neg R(c_x, c_y), \neg R(c_y, c_x), R(c_x, c_x)\}$ *($A1$ with $x \rightarrow c_x, y \rightarrow c_y, z \rightarrow c_x$)*

2. $\{\neg R(c_x, c_x)\}$ *($A2$ with $x \rightarrow c_x$)*

3. $\{\neg R(c_x, c_y), \neg R(c_y, c_x)\}$ *(resolve 1 and 2)*

4. $\{\neg R(c_y, c_x)\}$ *(resolve 3 and G1)*

5. $\emptyset$ *(resolve 5 and G2)*

$\Diamond$

4 Abstract Interpretation (5 points)

Consider the domain of integers intervals as a more concrete domain C, that is

$$C = \{\bot\} \cup \{[n, m] \mid n, m \in \mathbb{Z} \cup \{-\inf, +\inf\} \wedge n \leq m\}$$

and the domain of constant propagation as the abstract domain A, that is

$$A = \{\top, \bot\} \cup \{\{n\} \mid n \in \mathbb{Z}\}$$

Let $\alpha : C \to A$ and $\gamma : A \to C$ be given by

$$\alpha([n, n]) = \{n\}, \quad \alpha([n, m]) = \top \text{ for } n < m, \quad \alpha(\bot) = \bot$$

$$\gamma(\{n\}) = [n, n], \quad \gamma(\top) = [-inf, +inf], \quad \gamma(\bot) = \bot$$

Let $\leq$ denote the order on C where $\bot$ is the smallest element and otherwise $\leq$ is given by interval inclusion.

Let $\sqsubseteq$ denote the order on A where $\bot$ is the smallest element, $\top$ the largest, and all other elements are unrelated.

1. (1pt) Prove that α is monotonic. **Solution:** *Let $a, b \in C$ such that $a \leq b$. We proceed by case analysis. If $a = \bot$ then $\alpha(a) = \bot \sqsubseteq \alpha(b)$. If $a = b = [n, n]$ then $\alpha(a) = \alpha(b)$. If $b = [n, m]$ for $n < m$ then $\alpha(a) \sqsubseteq \top = \alpha(b)$. $\Diamond$*

2. (1pt) Prove that γ is monotonic. **Solution:** *Let $c, d \in C$ such that $c \sqsubseteq d$. We proceed by case analysis. If $c = \bot$ then $\gamma(c) = \bot \leq \gamma(d)$. We again proceed by case analysis. If $c = \bot$ then $\gamma(c) = \bot \leq \gamma(d)$. If $c = d = \{n\}$ then $\gamma(c) = \gamma(d)$. If $d = \top$ then $\gamma(c) \leq [-\inf, +\inf] = \gamma(d)$. $\Diamond$*

3. (1pt) Prove that the defining condition of the Galois connection holds. **Solution:** *We need to prove:*

$$\forall c \in C, a \in A.\ \alpha(c) \sqsubseteq a \iff c \leq \gamma(a)$$

The proof proceed again by case analysis on a and b. If $a = \bot$, $a = \top$, $c = \bot$ or $c = [-\inf, +\inf]$ then the condition holds trivially. Then, $\alpha([n, m]) \sqsubseteq \{x\} \iff x = n = m \iff [n, m] \leq \gamma(\{x\})$.
$\Diamond$

4. (1pt) Is function α injective? Prove or give a counterexample. **Solution:** *No. As counterexample, take* $\alpha([3, 14]) = \alpha([2, 71]) = \top$. $\Diamond$

5. (1pt) Is function γ injective? Prove or give a counterexample. **Solution:** *Yes. If* $\gamma(a) = \gamma(b) = \bot$ *then* $a = b = \bot$*. If* $\gamma(a) = \gamma(b) = [-\inf, +\inf]$ *then* $a = b = \top$*. If* $\gamma(a) = \gamma(b) = [n, n]$ *then* $a = b = \{n\}$*.* $[n, m]$ *is not in the range of* γ *for* $n < m$*.* $\Diamond$

5 Reachability (5 points)

Let $M = (S, I, r, A)$ be a transition system, $\bar{r} \subseteq S \times S$ defined as usual, and $G \subseteq S$ represent the set of good states. Let $D = 2^S \times 2^S$ (hence, D contains pairs of sets of states). Define $f : D \to D$ by

$$f((F, B)) = (F \cup sp(F, \bar{r}), B \cap wp(\bar{r}, B))$$

We will say that $(F, B) \in D$ is *safe* iff $F \subseteq B$. We will say that (F, B) is k-safe iff $f^k((F, B))$ is safe where we define, as usual,

$$f^0((F, B)) = (F, B), \qquad f^{n+1}((F, B)) = f(f^n((F, B)))$$

Your tasks:

1. (1pt) Define a partial order on D such that f is a monotonic function with respect to this order. **Solution:** *let $(F_1, B_1) \leq (F_2, B_2)$ iff $F_1 \subseteq F_2$ and $B_1 \supseteq B_2$. Monotonicity of f then follows from the monotonicity of sp and wp:*

   $$(F_1, B_1) \leq (F_2, B_2) \implies$$
   $$(F_1 \cup sp(F_1, \bar{r})) \subseteq (F_2 \cup sp(F_2, \bar{r})) \wedge (B_1 \cap wp(\bar{r}, B_1)) \supseteq (B_2 \cap wp(\bar{r}, B_2))$$

   Note: the way that this question is stated, there are other partial orders for which f is monotonic and that give a correct answer, though the order we show is the one we intended. ◊

2. (2pt) Show that if G is an invariant, then (I, G) is k-safe for all k.
   Hint: Let x and y be respectively members of the first and second element of $f^n((F, B))$. What can we state about traces of length n whose final state is x? What can we say about y? **Solution:** *We show by induction on k that if $f^k((I, G)) = (I', G')$ then $I' \subseteq R \subseteq G'$.*

   $k = 0$. Since G is an invariant, $I \subseteq R \subseteq G$. $k = n+1$. Assume $f((I, G)) = (I', G')$ is safe, so $I' \subseteq R \subseteq G'$. Then $f^{k+1}((I, G)) = f((I', G')) = (I' \cup sp(I', \bar{r}), G' \cap wp(\bar{r}, G'))$. Then,

   $$sp(I', \bar{r}) \subseteq R$$

   and

   $$R \subseteq wp(\bar{r}, G') \subseteq R$$

   and the conclusion follows. ◊

3. (2pt) Let n be the number of states in S. What is the smallest k as a function of n such that G is an invariant iff (I, G) is k-safe? **Solution:** *We will show that the answer is $\lfloor n/2 \rfloor$. Let I^k and G^k be respectively the first and second elements of $f^k((I, G))$, so that $f^k((I, G)) = (I^k, G^k)$. Then we can show by induction on k that:*

   - *I^k is the set of states reachable in at most k steps.*
   - *G^k is the set of states such that making up to k steps is guaranteed to stay inside G.*

   The condition $I^k \subseteq G^k$ then means that every path that has up to $2k$ steps ends in a state that is inside G: after first k steps we obtain a state in I^k, and if this state is in G^k then subsequent k steps remain inside G.

   If a system has n states, then the longest path has $n - 1$ steps. So it suffices to take as k the smallest integer such that $n - 1 \leq 2k$, in other words,

$$\lceil \frac{n-1}{2} \rceil = \lfloor \frac{n}{2} \rfloor$$

   $\Diamond$

6 Tseitin's Transformation (4 points)

Consider a propositional formula F in negation normal form, using only the connectives $\wedge, \vee, \neg$. Let the free variables of F be $p_1, \ldots, p_n$. Let $T(F)$ denote the result of Tseitin's transformation that replaces a subformula that is a conjunction or disjunction of literals $l_1 * l_2$ (where $*$ denotes "$\vee$" or "$\wedge$") by a new propositional variable q_i and introduces clauses corresponding to the propositional equivalence $q_i = l_1 * l_2$. Suppose that the free variables of $T(F)$ are $p_1, \ldots, p_n, q_1, \ldots, q_m$.

- (1pt) Given the best upper bound (over all possible F) on the number of assignments of $\{0, 1\}$ to variables $p_1, \ldots, p_n$ that make F true. **Solution:** 2^n. *To see that it is the best upper bound, consider the case where F is a tautology such as*

  $$p_1 \vee \neg p_1 \vee ... \vee p_n$$

  $\Diamond$

- (1pt) Give the best upper bound (over all possible F) on the number of assignments of $\{0, 1\}$ to variables $p_1, \ldots, p_n, q_1, \ldots, q_m$ that make $T(F)$ true. **Solution:** *We show a bijection between solutions to $T(F)$ and F. Consider the mapping that keps p_i unchanged and forgets the q_i. We need to show that it is injective and surjective.*

  Consider two satisfying assignements to $T(F)$. If the two solutions are identical on all p_i then they must be identical on all q_i as well, since the subformula $q_i = l_1 * l_2$ has to be satisfied in both assignemnts. Hence the mapping is injective. Moreover consider a satisfying assignment to F. We can construct a satisfying assignment to $T(F)$ by setting $q_i = l_1 * l_2$ to true if and only if $l_1 * l_2$ evaluates to true, so our mapping is surjective.

  Therefore, the number of satisfying assignments to $T(F)$ is the same as the number of satisfying assignments to F, 2^n. $\Diamond$

- (1pt) There is a variant T_1 of Tseitin's transformation that introduces clauses corresponding to an implication, instead of the propositional equivalence $q_i = (l_1 * l_2)$ and still generates an equisatisfiable formula. What is the direction of the implication that should be used for the introduced clauses, $q_i \rightarrow (l_1 * l_2)$ or $(l_1 * l_2) \rightarrow q_i$? **Solution:** *Consider the formula $F := p_1 \wedge (\neg p_1)$,* *which is unsatisfiable. Then* $T(F) = q_1 \wedge (q_1 \rightarrow (p_1 \wedge (\neg p_1)))$ *is equisatisfiable. If we used the other direction, we would get* $q_1 \wedge ((p_1 \wedge (\neg p_1)) \rightarrow q_1)$ *which is satisfiable by setting* q_1 *to true, and hence not equisatisfiable with* F. $\Diamond$

- (1pt) Give an upper bound on the number of satisfying assignments for the result of such optimized Tseitin's transformation assuming it generates a formula $T_1(F)$ with variables $p_1, \ldots, p_n, q_1, \ldots q_m$. **Solution:** 2^{n+m} *is an upper bound.* $\Diamond$

7 Predicate Abstraction (6 points)

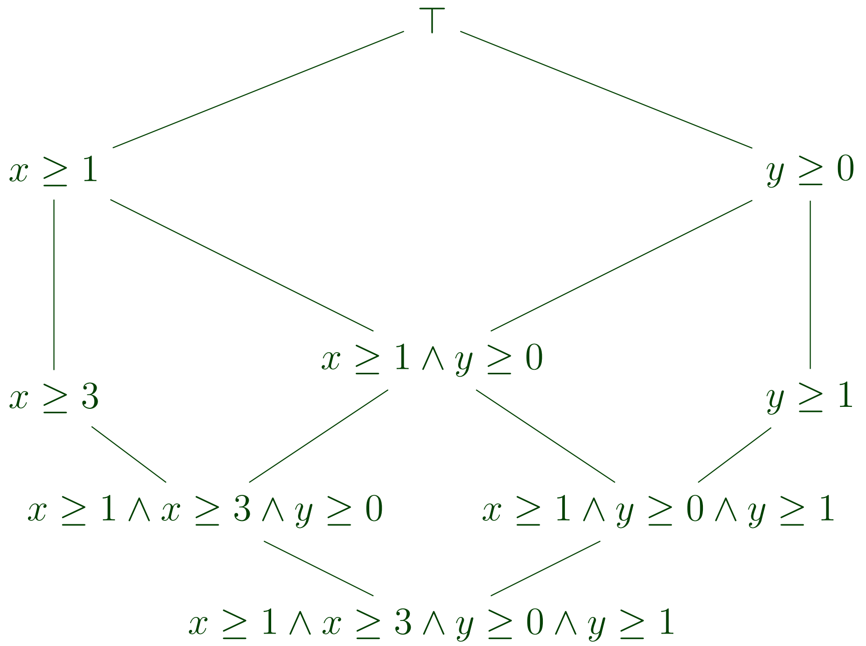
Consider the following program over mathematical (unbounded) integers:

```
def myLittleProgram(var x: BigInt, var y: BigInt): Int =
  require(x >= 1 && y >= 1) ⓪
  x = x + y
  while ① (x >= y + 3) {
    ②
    if x % 2 == 0 then ③
      y = y + 1 ④
      x = x / 2 − 1 ⑤
    else ⑥
      x = x + 1 ⑦
      y = 0 ⑧
  }
  ⑨ y = x ∗ y
  ⑩ 1 / y
```

Note that / denotes integer division, rounded down. We identify program points using numbers ⓪,①,②,... Consider this set of 6 predicates:

$$\mathcal{P} = \{x \geq 1, x \geq 3, y \geq 0, y \geq 1, (y \geq 1 \vee x \geq 3), x\%2 = 0\}$$

1. (1pt) Let $\mathcal{P}' \subseteq \mathcal{P} = \{x \geq 1, x \geq 3, y \geq 0, y \geq 1\}$. Draw the Hasse diagram for the lattice $2^{\mathcal{P}'}$ of conjuncts over $\mathcal{P}'$ ordered by the predicate abstraction lattice order. **Solution:** *Note that $\top$ is the empty conjunction (corresponding to the empty subset of $\mathcal{P}'$). The Hasse diagram is:*


$\top$
$x \geq 1$
$y \geq 0$
$x \geq 1 \wedge y \geq 0$
$x \geq 3$
$y \geq 1$
$x \geq 1 \wedge x \geq 3 \wedge y \geq 0$
$x \geq 1 \wedge y \geq 0 \wedge y \geq 1$
$x \geq 1 \wedge x \geq 3 \wedge y \geq 0 \wedge y \geq 1$

$\Diamond$

2. (1pt) Note that we did not include the predicate "false" in $\mathcal{P}$, but predicate abstraction should still work. Let the abstraction function from sets of states to sets of predicates to be α. Compute $\alpha(\emptyset)$ and $\gamma(\alpha(\emptyset))$.

   Solution:

   $$\alpha(\emptyset) = x \geq 1 \wedge x \geq 3 \wedge y \geq 0 \wedge y \geq 1 \wedge (y \geq 1 \vee x \geq 3) \wedge x\%2 = 0$$

   (i.e. the conjunction of all predicates in $\mathcal{P}$).

   $$\gamma(\alpha(\emptyset)) = \{(x, y) \in \mathbb{N}^2 \mid x \geq 3 \wedge y \geq 1 \wedge x\%2 = 0\}$$

   . $\Diamond$

3. (4pt) Using predicate abstraction as abstract interpretation, compute, for each indicated program point of myLittleProgram, the set of predicates from $\mathcal{P}$ that hold at that point. Only show the final conjunction of predicates for each point, not the entire iterative process.
   Does the annotation before the last line contain the predicate $y \geq 1$? **Solution:**

   - ⓪ $x >= 1 \wedge y >= 1$

- ① $y >= 0 \wedge x >= 1 \wedge (y >= 1 || x >= 3)$
- ② $y >= 0 \wedge x >= 3$
- ③ $y >= 0 \wedge x >= 3 \wedge x\%2 = 0$
- ④ $y >= 1 \wedge x >= 3 \wedge x\%2 = 0$
- ⑤ $y >= 1 \wedge x >= 1$
- ⑥ $y >= 0 \wedge x >= 3$
- ⑦ $y >= 0 \wedge x >= 3$
- ⑧ $y >= 0 \wedge x >= 3$
- ⑨ $y >= 1 \wedge x >= 1$
- ⑩ $y >= 1 \wedge x >= 1$

$\Diamond$