EPFL Formal Verification Course Exam, 23 November 2023

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).

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 one, A4-sized, two-sided cheat sheet.

The maximal number of points on the exam is 40. We advise you to first solve questions that you find easier. You can use reasonably high-level mathematical proofs except in part 4.3 where we will need an explicit sequence of proof steps. 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:

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

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 Transition Systems and Invariants (8pt)

Let $M = (S, I, r, A)$ be a transition system with $I \subseteq S$ and $I \neq \emptyset$ a non-empty initial set of initial states, $r \subseteq S \times A \times S$ the transition relation and A the (non empty) input alphabet.

Part I. For each of the following, prove or give a counter-example.

1.1 Diagonal

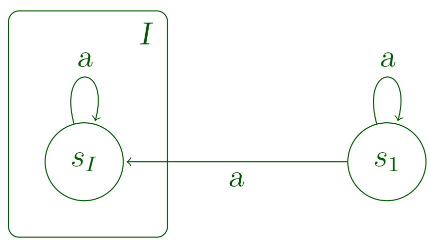
If $\bar{r}$ is the diagonal relation, then every subset of S is an inductive invariant.

Solution: *This is* ***false****. The empty set is a subset of S, yet it is not an inductive invariant.*
$\Diamond$

1.2 Lurking Havoc

If there exists an $s_1 \in S$ such that for all $s_2 \in S$, $(s_1, s_2) \in \bar{r}$, then the smallest inductive invariant is S.

Solution: *This is also* ***false****. If s_1 is not reachable then we can have smaller inductive invariants. In the following example, $\{s_I\}$ is an inductive invariant, yet it is a strict subset of S. For the statement to hold, s_1 should belong to $Reach(M)$.*



$\Diamond$

1.3 Rock Star

For any s_2: if for all $s_1 \in S$ we have $(s_1, s_2) \in \bar{r}$, then $s_2 \in Reach(M)$. Stated as a formula:

$$\forall s_2. ((\forall s_1 \in S.(s_1, s_2) \in \bar{r}) \longrightarrow s_2 \in Reach(M))$$

Solution: *This is* ***true****. Since $I \neq \emptyset$, there exists a $s_I \in I$. For an arbitrary s_2, if the left part of the implication holds, then it holds in particular for s_I and we have $(s_I, s_2) \in \bar{r}$. Hence:*

$$s_2 \in \bar{r}[I] \subseteq \bar{r}^*[I] = Reach(M)$$

$\Diamond$

1.4 Transitions over a Lattice

If S is a complete lattice with the order relation $\leq$ and with the greatest lower bound of a set denoted by $\bigsqcap$, then if for all s_1, s_2, $(s_1, s_2) \in \bar{r} \to s_1 \leq s_2$, then for $u = \bigsqcap I$, the set $I_u = \{s \in S \mid u \leq s\}$ is an inductive invariant.

Solution: *This is also **true**. Since any element of a set is greater or equal than its glb, $I \subseteq I_u$. If $s \in I_u$ and $(s, s') \in \bar{r}$, then $u \leq s \leq s' \implies u \leq s' \implies s' \in I_u$. Therefore I_u is an inductive invariant.* $\Diamond$

1.5 No Junk

If, for all $s \in S \setminus I$, $\bar{r}[\{s\}] \subseteq Reach(M)$, then every invariant is an inductive invariant.

Solution: *This is **true**. Let $\mathrm{Inv} \supseteq Reach(M)$, an invariant of M. By definition of $Reach(M)$, we know that for all $s \in I$, $\bar{r}[\{s\}] \subseteq Reach(M)$. If additionaly, for any $s \in S \setminus I$, $\bar{r}[\{s\}] \subseteq Reach(M)$, then for any $(s, s') \in \bar{r}$ with $s \in \mathrm{Inv}$ we also have $s' \in \mathrm{Inv}$. This means that Inv is an inductive invariant.* $\Diamond$

****

Part II. In the following parts, for any set of initial states $I' \subseteq S$ and relation $r' \subseteq S \times A \times S$, let $M(I', r')$ denote the transition system (S, I', r', A) and let $Reach(I', r')$ denote $Reach(M(I', r'))$. For each of the following, prove or give a counter-example.

1.6 One Reach, Two Reaches

For any r_1, r_2 and I, $Reach(Reach(I, r_1), r_2) \subseteq Reach(I, r_1 \cup r_2)$.

Solution: *This is also **true**:*

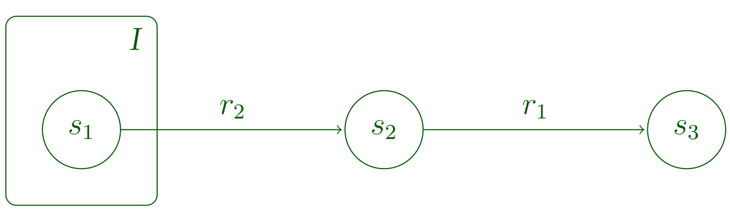
$$\begin{aligned} Reach(Reach(I, r_1), r_2) &\subseteq Reach(Reach(I, r_1 \cup r_2), r_2) \\ &\subseteq Reach(Reach(I, r_1 \cup r_2), r_1 \cup r_2) \\ &= Reach(I, r_1 \cup r_2) \end{aligned}$$

$\Diamond$

1.7 Both Reach

For any r_1, r_2 and I, $Reach(I, r_1 \cup r_2) \subseteq Reach(Reach(I, r_1), r_2)$.

Solution: *Counterexample:*



$\Diamond$

1.8 Invariant Part by Part

For any r_1, r_2 and I, if i_1 is an invariant of $M(I, r_1)$ and i_2 is an invariant of $M(I, r_2)$ then $i_1 \cup i_2$ is an invariant of $M(I, r_1 \cup r_2)$.

Solution: *This is **false** as the previous counterexample also works here. $\{s_1\}$ is an invariant of $M(I, r_1)$ and $\{s_1, s_2\}$ of $M(I, r_2)$. However, their union is not an invariant of $M(I, r_1 \cup r_2)$.*
$\Diamond$

More questions on next pages!

2 Quantifier Elimination (5pt)

These problems can be solved using the technique of quantifier elimination for linear arithmetic that we studied in the lectures. You can skip steps when transforming quantifier-free sub-formulas into equivalent ones. You can use your own strategy for applying quantifier elimination steps.

2.1 An Integer Formula

Consider the formula $F(x, z)$ where the variables range over the mathematical integers $\mathbb{Z}$:

$$\forall y.((x < y \land y < z) \longrightarrow (3 \mid (y+1) \lor 3 \mid (y+2)))$$

($3 \mid (y+1)$ stands for "3 divides $(y+1)$".)
Find a quantifier-free formula equivalent to $F(x, z)$.

Solution:

$$\begin{aligned}
&\forall y.((x < y \land y < z) \longrightarrow (3 \mid (y+1) \lor 3 \mid (y+2))) \\
\equiv &\neg \exists y.(x < y \land y < z \land \neg(3 \mid (y+1)) \land \neg(3 \mid (y+2))) \\
\equiv &\neg \exists y.(x < y \land y < z \land 3 \mid y) \qquad \textit{since 3 divides either } y,\ y+1 \textit{ or } y+2 \\
\equiv &\neg \left(\bigvee_{i=1}^{3} x + i < z \land 3 \mid x + i \right)
\end{aligned}$$

◊

2.2 A Rational Arithmetic Formula

Consider the following formula $G(x, z)$ where the variables range over rational numbers $\mathbb{Q}$:

$$\forall y.((x < y \land y < z) \longrightarrow \forall u.(x \neq u + u + u))$$

Find a quantifier-free formula equivalent to $G(x, z)$.

Solution:

$$\begin{aligned}
&\forall y.((x < y \land y < z) \longrightarrow \forall u.(x \neq u + u + u)) \\
\equiv &\neg \exists y.(x < y \land y < z \land \exists u.(x = u + u + u)) \\
\equiv &\neg \exists y.(x < y \land y < z) \\
\equiv &\neg (x < z)
\end{aligned}$$

◊

More questions on next pages!

3 Programs and Formulas (9pt)

Consider a program with two variables ranging over unbounded integers $\mathbb{Z}$, so let $S = \mathbb{Z}^2$ be the set of possible states of the program at a program point.

3.1 Program to Formula

Convert the following program into a formula that expresses the program's meaning:

```
x= x + y
if x > y then
  x= x - y
else
  y= y - x
```

Your formula should have:

1. x, y, x', y' as the only *free* variables
2. "$\exists$" as the only quantifier (no universal quantifiers); it may also be quantifier-free
3. no relations or functions other than those of integer linear arithmetic.

Solution:

- *x = x + y command:* $F_1 = x' = x + y \wedge y' = y$
- ***if then else** command:* $F_2 = (x > y \wedge x' = x - y) \vee (x \leq y \wedge y' = y - x)$
- *Program formula:*

$$\exists x'', y''. F_1[x' := x'', y' := y''] \wedge F_2[x := x'', y := y'']$$

$$\equiv \exists x'', y''.\ x'' = x + y \ \wedge\ y'' = y \ \wedge\ ((x'' > y'' \wedge x' = x'' - y'') \vee (x'' \leq y'' \wedge y' = y'' - x''))$$

$\Diamond$

3.2 About That Program

Let $r \subseteq S \times S$ be the semantics of the program in the previous part. Consider the initial set of states p:

$$p = \{(x, y) \in S \mid 0 \leq x + y\}$$

Find the formula that describes the strongest postcondition of p with respect to r. Specifically, compute a quantifier-free formula Q containing as the variables only x and y, such that the set

$$q = \{(x, y) \mid Q\}$$

is the relation image of set p under relation r, that is, $q = r[p]$ holds.

You do not need to show detailed steps, but be careful to give a formula for the exact relation image (strongest postcondition).

3.3 Quadratic Mess

This part is solvable independently of the previous two. Consider the initial set of states

$$p = \{(x, y) \in S \mid x^2 \le y\}$$

Let $r \subseteq S \times S$ now be the meaning of the following program with two assignments executed one after another:

```
y= y - 1
x= x + y
```

Compute a quantifier-free formula Q containing as variables only x and y that characterizes the strongest postcondition of p with respect to r, that is, a formula Q such that the set

$$q = \{(x, y) \mid Q\}$$

is the relation image of set p under relation r, that is, $q = r[p]$ holds.

Solution: *It is quite immediate that y' can range in $[-1, \infty)$ since y has to be nonnegative. Given y, we deduce from the constraint that $-\sqrt{y'+1} \le x \le \sqrt{y'+1}$. Since $x' = x + y'$, x' has to be bounded between $y' - \sqrt{y'+1}$ and $y' + \sqrt{y'+1}$. Therefore*

$$r[p] = \{(x', y') \in S \mid y' \ge -1 \land y' - \sqrt{y'+1} \le x' \le y' + \sqrt{y'+1}\}$$

$\Diamond$

More questions on next pages!

4 First Order Logic Resolution (7pt)

Consider the following first order logic signature $L = \{E, a, f, g\}$, consisting of a binary relation symbol E, a constant a, a function f taking 1 argument, and a function g taking 1 argument. Recall that, in FOL, we distinguish terms (denoting values in the domain) from formulas (denoting truth values).

4.1 Defining Terms

Let V denote a countable set of variables used to build terms and formulas. Let T denote the set of all terms with variables V in the signature L. Let GT denote the set of ground terms corresponding in the signature L (domain of the Herbrand universe).

a) Does $g(f(a))$ belong to GT?

b) Does $g(f(x))$ belong to GT?

c) Does $E(f(a), a)$ belong to GT?

d) Does $E(f(x), a)$ belong to T?

e) Give a definition for a function $H : 2^T \to 2^T$ such that these two conditions hold:

$$T = \bigcup_{i \geq 0} H^i(V)$$

$$\mathsf{GT} = \bigcup_{i \geq 0} H^i(\emptyset)$$

Do not use GT in the definition of H. (This way, we can use H to define GT.)

($H^i(x)$ denotes the iterated application of H, i.e., $H^0(x) = x$, $H^{i+1}(x) = H(H^i(x))$.)

Solution: *Recall from the lecture:*
GT is the least set such that if a function or constant symbol h is in L, $ar(h) = n$ and $t_1, \ldots, t_n \in \mathsf{GT}$ then $h(t_1, \ldots, t_n) \in \mathsf{GT}$.

Therefore we have $g(f(a)) \in \mathsf{GT}$, but not $g(f(x))$, $E(f(a), a)$, $E(f(x), a)$ as they contain symbols that are not in L or relation symbols.

The function H is defined as follow:

$$H(S) = S \cup \{a\} \cup \{f(s) \mid t \in S\} \cup \{g(s) \mid t \in S\}$$

(It was not asked to justify your answer)
When the argument given to H^i is the empty set, we obtain the same inductive definition

for the set of ground terms than in the lecture.
The terms of the language are

$$t ::= x \mid a \mid f(t) \mid g(t)$$

That is, the least fixpoint of the function $\mathrm{Terms} : S \mapsto V \cup \{a\} \cup \{f(s) \mid t \in S\} \cup \{g(s) \mid t \in S\}$. *Since* Terms *is monotonic and* ω*-continuous, its least fixpoint is* $\mathrm{Terms}^*(\emptyset)$. *We therefore need to prove that* $H^*(V) = \mathrm{Terms}^*(\emptyset)$. *First note that if* $V \subseteq S$, $H(S) = \mathrm{Terms}(S) \cup S$. *We then prove by induction that for all i,* $H^i(V) = \mathrm{Terms}^i(\emptyset)$. *The base case is immediate. Induction step:*

$$\begin{aligned}
\bigcup_{i \le n+1} H^i(V) &= \bigcup_{i \le n} \mathrm{Terms}^i(\emptyset) \cup H(H^n(V)) \\
&= \bigcup_{i \le n} \mathrm{Terms}^i(\emptyset) \cup H\left(\bigcup_{i \le n} H^i(V)\right) \\
&= \bigcup_{i \le n} \mathrm{Terms}^i(\emptyset) \cup \mathrm{Terms}\left(\bigcup_{i \le n} H^i(V)\right) \cup \bigcup_{i \le n} H^i(V) \\
&= \bigcup_{i \le n} \mathrm{Terms}^i(\emptyset) \cup \mathrm{Terms}\left(\bigcup_{i \le n} \mathrm{Terms}^i(\emptyset)\right) \cup \bigcup_{i \le n} \mathrm{Terms}^i(\emptyset) \\
&= \bigcup_{i \le n} \mathrm{Terms}^i(\emptyset) \cup \bigcup_{i \le n} \mathrm{Terms}^{i+1}(\emptyset) \\
&= \bigcup_{i \le n+1} \mathrm{Terms}^i(\emptyset)
\end{aligned}$$

Therefore $H^*(V) = \mathrm{Terms}^*(\emptyset) = T$.

$\Diamond$

4.2 Axioms and Their Normal Form

Consider the following formula A:

$$\begin{aligned}
\forall x, y, z. \qquad & (E(x,y) \wedge E(y,z) \to E(x,z)) \wedge \\
& (E(x,y) \to (E(f(x), f(y)) \wedge E(g(x), g(y)))) \wedge \\
& E(f(g(x)), g(f(x)))
\end{aligned}$$

Show the result of transforming the above formula into an *equivalent* finite set of first-order clauses.

Solution:

$$\textbf{\textit{NNF:}} \quad \begin{array}{ll} \forall x,y,z. & (\neg E(x,y) \vee \neg E(y,z) \vee E(x,z)) \wedge \\ & (\neg E(x,y) \vee (E(f(x),f(y)) \wedge E(g(x),g(y)))) \wedge \\ & E(f(g(x)),g(f(x))) \end{array}$$

$$\textbf{\textit{PNF:}} \quad \begin{array}{l} (\neg E(x,y) \vee \neg E(y,z) \vee E(x,z)) \wedge \\ (\neg E(x,y) \vee (E(f(x),f(y)) \wedge E(g(x),g(y)))) \wedge \\ E(f(g(x)),g(f(x))) \end{array}$$

$$\textbf{\textit{CNF:}} \quad \begin{array}{l} (\neg E(x,y) \vee \neg E(y,z) \vee E(x,z)) \wedge \\ (\neg E(x,y) \vee E(f(x),f(y))) \wedge \\ (\neg E(x,y) \vee E(g(x),g(y))) \wedge \\ E(f(g(x)),g(f(x))) \end{array}$$

$$\textbf{\textit{Clauses:}} \quad \begin{array}{l} \{\{\neg E(x,y), \neg E(y,z), E(x,z)\}, \\ \{\neg E(x,y), E(f(x),f(y))\}, \\ \{\neg E(x,y), E(g(x),g(y))\}, \\ \{E(f(g(x)),g(f(x)))\}\} \end{array}$$

$\Diamond$

4.3 Applying Resolution

Use the clauses obtained in the previous part to show that $E(f(f(g(a))), g(f(f(a))))$ is a consequence of formula A.

Use a refutation proof with the rule of FOL resolution with instantiation. Write your proof as a numbered sequence proving the empty clause $\emptyset$. For each step indicate if it is an assumption or write "from n_1, n_2" where n_1 and n_2 are previous steps from which it follows.

You may abbreviate the terms using prefix notation, writing e.g. $E(f(f(g(a))), g(f(f(a))))$ as $E(\text{ffga}, \text{gffa})$.

Solution: *Let's first prove the statement in a more classical way to get an intuition on which step should be used in the resolution proof. We know that $E(f(g(a)), g(f(a)))$ and $E(f(g(f(a))), g(f(f(a))))$. From the former, we deduce $E(f(f(g(a))), f(g(f(a))))$ and hence $E(f(f(g(a))), g(f(f(a))))$.*

Let's now write the resolution proof:

Ax 1 $\{\neg E(x,y), \neg E(y,z), E(x,z)\}$

Ax 2 $\{\neg E(x,y), E(f(x),f(y))\}$

Ax 3 $\{\neg E(x,y), E(g(x),g(y))\}$

Ax 4 $\{E(f(g(x)),g(f(x)))\}$

1. $\{\neg E(\text{ffga}, \text{gffa})\}$
2. $\{E(\text{fga}, \text{gfa})\}$ *by instantiation of Ax 4 with* $x := a$

3. $\{\neg E(fga, gfa), E(ffga, fgfa)\}$ *by instantiation of Ax 2 with* $x := fga$ *and* $y := gfa$

4. $\{E(ffga, fgfa)\}$ *by 2 and 3*

5. $\{E(fgfa, gffa)\}$ *by instantiation of Ax 4 with* $x := fa$

6. $\{\neg E(ffga, fgfa), \neg E(fgfa, gffa), E(ffga, gffa)\}$ *by instantiation of Ax 1 with* $x := ffga$, $y := fgfa$ *and* $z := gffa$

7. $\{\neg E(fgfa, gffa), E(ffga, gffa)\}$ *by 4 and 6*

8. $\{E(ffga, gffa)\}$ *by 5 and 7*

9. $\emptyset$ *by 1 and 8*

$\Diamond$

5 The Age of AI - Abstract Interpretation (11pt)

In this question we are designing an abstract domain that improves on intervals by tracking divisibility as well. Even in case your understanding of abstract interpretation is limited, your intuition and understanding of Hoare triples and strongest postconditions may allow you to solve some of these problems.

For simplicity, consider programs with a single variable. A set of states is a subset of the set of integers, so the concrete domain is the set of all subsets, $C = 2^{\mathbb{Z}}$. The ordering on concrete elements is $\subseteq$, which gives a (complete) lattice with the least upper bound $\cup$ and the greatest lower bound $\cap$.

The abstract domain elements are four-tuples (a, b, c, d) where a, b, c, d can be integers and where a can also be $-\infty$ and b can also be $+\infty$. We assume that $-\infty \leq x$ and $x \leq +\infty$ for every $x \in \mathbb{Z}$. Hence,

$$A = \{(a, b, c, d) \mid a \in \{-\infty\} \cup \mathbb{Z}, b \in \mathbb{Z} \cup \{+\infty\}, c, d \in \mathbb{Z}\}$$

Define

$$\gamma(a, b, c, d) = \{x \mid a \leq x \leq b \land \exists k \in \mathbb{Z}.\ x = kd + c\}$$

5.1 Special Case

Give a simple definition of the set $\gamma(a, b, 0, 1)$ and for the set $\gamma(a, a, 0, 1)$. **Solution:**

$$\gamma(a, b, 0, 1) = [a, b] \text{ and } \gamma(a, a, 0, 1) = \{a\}$$

$\Diamond$

5.2 Abstract Strongest Postcondition for Assignment

Consider the following assignment statement $c1$:

```
x = x - 3
```

Let $r_1 \subseteq \mathbb{Z}^2$ be the meaning of that statement. Write down an expression defining r_1.

Then, give a definition of a function $F_1^\# : A \to A$ such that, for all $x \in A$,

$$r_1[\gamma(x)] \subseteq \gamma(F_1^\#(x)) \tag{1}$$

Try to define $F_1^\#$ such that $\gamma(F^\#(x))$ is as small set as possible while satisfying the above condition.

Illustrate your definition by showing and simplifying the mathematical expression for $F^\#((a, a, 0, 1))$ where $a \in \mathbb{Z}$.

Solution:

$$F_1^\#((a, b, c, d)) = (a - 3, b - 3, c - 3, d)$$

Indeed

$$\begin{aligned} & a \leq x \leq b \wedge \exists k \in \mathbb{Z}.\ x = kd + c \\ \iff & a - 3 \leq x - 3 \leq b - 3 \wedge \exists k \in \mathbb{Z}.\ x - 3 = kd + c - 3 \end{aligned}$$

◊

5.3 Abstract Strongest Postcondition for Tests

Analogously to the previous part, consider the command $c2$:

```
assume(x > 3)
```

whose meaning is relation r_2. Give functions $F_2^\# : A \to A$ that satisfies the condition analogous to (1):

$$r_2[\gamma(x)] \subseteq \gamma(F_2^\#(x))$$

Also give $F_3^\# : A \to A$ that corresponds to the command:

```
assume(x <= 3)
```

Solution: *Let* $x = (a, b, c, d)$

$$\begin{aligned} r[\gamma(x)] &= \{x \mid x > 3\} \cap \{x \mid a \leq x \leq b \wedge \exists k \in \mathbb{Z}.\ x = kd + c\} \\ &= \{x \mid x \leq 4\} \cap \{x \mid a \leq x \leq b \wedge \exists k \in \mathbb{Z}.\ x = kd + c\} \\ &= \{x \mid \max(4, a) \leq x \leq b \wedge \exists k \in \mathbb{Z}.\ x = kd + c\} \end{aligned}$$

Thus,

$$F_2^\#((a, b, c, d)) = (\max(4, a), b, c, d)$$

Similarly,

$$F_3^\#((a, b, c, d)) = (a, \min(3, b), c, d)$$

◊

5.4 Joining

Propose a definition of $J : A \times A \to A$ such that for all x, y:

- $J(x, y) = J(y, x)$
- $\gamma(x) \subseteq \gamma(J(x, y))$

and such that $\gamma(J(x, y))$ is as small as you can make it while satisfying the above two conditions.

Solution:
Let $(a_1, b_1, c_1, d_1), (a_2, b_2, c_2, d_2) \in A$ *and* $d' = \gcd(d_1, d_2, |c_2 - c_1|)$, $c' = \min(c_1, c_2) \;\%\; d'$.

$$J((a_1, b_1, c_1, d_1), (a_2, b_2, c_2, d_2)) = (\min(a_1, a_2), \max(b_1, b_2), c', d')$$

We show that if $x \in \gamma((a_1, b_1, c_1, d_1))$, *then* $x \in \gamma(J((a_1, b_1, c_1, d_1), (a_2, b_2, c_2, d_2)))$. *The most technical part consists in proving that if* $x = k_1 d_1 + c_1$ *for some* k_1 *then there is a* k_2 *such that* $x = k_2 d' + c'$. *We know that* d' *divides* d_1 *and* $\min(c_1, c_2) = d' k_3 + c'$ *for some* k_3. *Since* d' *divides* $|c_2 - c_1|$, *we have* $c_1 = d' k_4 + c'$ *for some* k_4. *Therefore*

$$\begin{aligned} x &= k_1 d_1 + c_1 \\ &= k_1 k_5 d' + d' k_4 + c' \\ &= (k_1 k_5 + k_4) d' + c' \end{aligned}$$

$\Diamond$

5.5 Loop and Its Control-Flow Graph

You may be able to solve this part independently of the other parts.

Consider the following small program.

```
// 1
x = 20
while // 2
        x > 3 do
        // 3
      x = x - 3
// 4
```

Draw a control flow diagram with these 4 program points, with edges labeled by assignments and tests ("assume" statements).

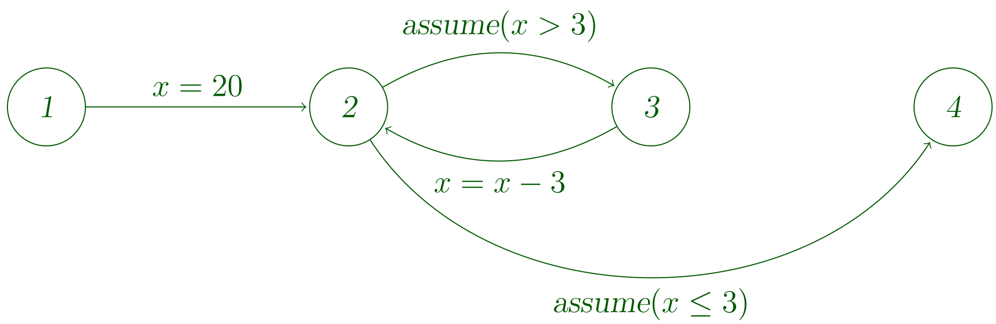
Let I_1 be the mathematical formula "true".

Find formulas I_2, I_3, I_4 expressible in the language of integer linear arithmetic with the divisibility operator such that I_1, I_2, I_3, I_4 result in valid Hoare triples according to the control flow graph, and such that they are as strong as you can make them (making all of them "true" is not a good solution).

For example, the following should be valid Hoare triples (among others):

- I_1 {`x = 20`} I_2
- I_2 {`assume(x > 3)`} I_3
- I_3 {`x = x - 3`} I_2

Solution:



$$
\begin{aligned}
I_1 &:= \top \\
I_2 &:= 3 \mid x + 1 \\
I_2 &:= x > 3 \wedge 3 \mid x + 1 \\
I_4 &:= x = 2
\end{aligned}
$$

$\Diamond$

5.6 Injectivity

Give an example showing that γ is not injective.
Define a subset $A_N \subseteq A$ such that γ restricted to A_N is injective and $\gamma[A_N] = \gamma[A]$.
We write γ_N for the restriction of γ to A_N: $\gamma_N(x) = \gamma(x)$ for all $x \in A_N$.

Solution: $\gamma((1, 0, 1, 0)) = \gamma((2, 0, 1, 0)) = \emptyset$ *meaning that* γ *is not injective. We start by restraining ourselves to a unique representation for divisibility with* $d > 0$ *and* $c < d$*. We also want* a *and* b *to be divisible by* d *so that they also belong to the set. We want to keep only one representation for the empty set. Therefore,* a *should be smaller than* b *and* $[\infty, \infty]$ *should be removed as well. If* $a = b = -\infty$*, we set by convention that* $c = d = 0$*. Finally if the set is a singleton, we choose* $c = 0$ *and* $d = a$*. We end up with the following subset of* A

$$
\begin{aligned}
A_N = &\{(a, b, c, d) \mid a \in \mathbb{Z} \wedge b \in \mathbb{Z} \wedge a < b \wedge d \mid a \wedge d \mid b \wedge d > 0 \wedge c < d\} \\
&\cup \{(-\infty, b, c, d) \mid b \in \mathbb{Z} \wedge d \mid b \wedge d > 0 \wedge c < d\} \\
&\cup \{(a, \infty, c, d) \mid a \in \mathbb{Z} \wedge d \mid a \wedge d > 0 \wedge c < d\} \\
&\cup \{(-\infty, \infty, c, d) \mid d > 0 \wedge c < d\} \\
&\cup \{(a, a, 0, a) \mid a \in \mathbb{Z}\} \\
&\cup \{(-\infty, -\infty, 0, 0)\}
\end{aligned}
$$

$\Diamond$

5.7 Ordering on A

Define a binary partial order relation $\leq$ on A_N such that γ_N is an injective monotonic function from A_N to C.

Solution:

$$(a_1, b_1, c_1, d_1) \leq (a_2, b_2, c_2, d_2) \iff (a_2 \leq a_1 \wedge b_1 \leq b_2 \wedge c_2 = c_1 \bmod d_2 \wedge d_2 \mid d_1)$$

In particular, with this definition, $(a, b, 0, 1) \geq (a, b, c, d)$. $\Diamond$

5.8 Galois Insertion

Can you define $\alpha : C \to A_N$ such that (α, γ_N) form a Galois insertion between C and A_N? (Reminder: a Galois insertion is a Galois connection where γ_N is injective.)

Solution: *Let* $x \in C$ *and let* $d = \gcd(\{|e_1 - e_2| \mid e_1, e_2 \in x\})$. *If* x *is a singleton, set* d *to the only value in the set. To compute* c*, take any element of* x *and compute the remainder modulo* d*. We then define*

$$\alpha(x) = (\min x, \max x, c, d)$$

If $x = \emptyset$ *then* $\alpha(x) = (-\infty, -\infty, 0, 0)$. *We have*

$$\alpha(\gamma_N(a)) = a \text{ and } \gamma_N(\alpha(x)) \supseteq x$$

Indeed, with the definition of d*, we choose the smallest possible step. Therefore, performing* $\alpha(x)$ *adds missing elements such that we have an element in the set for every multiple of* d *(plus* c*).* $\Diamond$

End of the Exam Sheet