# Software Verification

Monday, January 10th 2022, 3 hours

This assignment contains three independent parts: the first part deals with bounded model-checking, the second part is about abstract interpretation, and the last part addresses algorithmic representations of upward-closed sets.

**All documents are authorized during the examination**
**Answers can be written in French**

## 1 Bounded Model-Checking (12pts)

This section will browse a few concepts that have been seen in the first part of the course (Bounded Model-Checking).

### 1.1 Principles

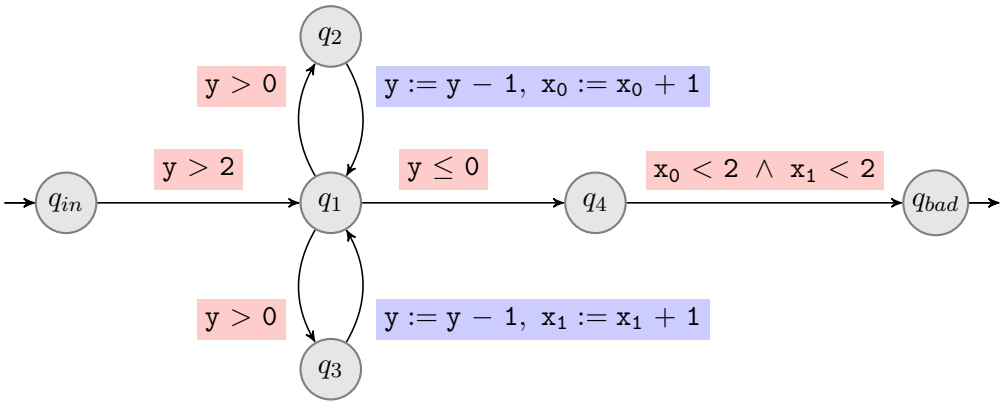
We start with some questions to check your understandings of the Bounded Model-Checking algorithms. We recall that given a transition system $\mathcal{T}$, the Bounded Model-Checking at depth $n$ explore all executions of length at most $n$ and check if one of them reach an undesirable state (such an execution is called *faulty*).

**Question 1** *A Bounded Model-Checking search can return three answers: Counter-example found, No counter-example found (exhaustive), No counter-example found (non-exhaustive).*

*For each possible answer, explain what the answer means about the presence of a faulty execution, its size, and the size of all executions of the model.*

**Question 2** *We have seen in course two algorithm : Depth-First Search, and Global. Recall briefly how the executions are explored in the two algorithms and what guarantee they give on the faulty execution they return if they return one.*

### 1.2 Executing on an example



Observe that in the system depicted above, we have allowed to do two affectations in parallel or to test two conditions at the same time (to shorten the model for making the paper exploration easier). You will of course use formulæ that do the two at the same time.

The state $q_4$ is considered to be the ending state of the program, the transition from it to $q_{bad}$ is the specification of the program. $q_{bad}$ is of course the error state that we want to test if it is reachable or not.

**Question 3** *Give an intuition on what the system is achieving.*

**Question 4** *Apply the bounded model-checking algorithm to the previous system up to bound* 6*: draw the exploration tree displaying the formulæ added (only the added one not the complete formula) at each step in the Depth-First Search algorithm, highlight the unsatisfiable ones, and give the result of the algorithm.*

**Question 5** *Is there a bound on which the algorithm would give a different answer? Justify (either explain why the algorithm will always answer correct, or give a failing execution).*

**Question 6** *If there is a bug, propose a simple modification (you cannot change the transition from $q_4$ to $q_{bad}$, as we want the same specification) that will render the system correct. We want the informal idea of the algorithm to stay the same as well, i.e., keep the transitions already existing. Another note: if we replace $y > 2$ by $y > 1$ in the transition from $q_{in}$ to $q_1$, the system should be incorrect (in particular, fixing the values of the $x$ in $q_4$ is not acceptable).*

*Will the bounded model checking terminate on that new system (or the old, in case there is no bug), both in forward and backward ? If not, can you give a path of length at least $k$ for any $k$ ?*

### 1.3 The Boolean Case

Historically, the Bounded Model-Checking was first developed for hardware verification, on which variables are boolean. In the following, we consider such a system with $M$ states and $n$ boolean variables $x_1, \ldots, x_n$.

**Question 7** *Prove that in that setting, the Bounded Model-Checking is complete, i.e. there exists a bound $k$ for which if the algorithm has not found a counter-example, we know there is none, even for longer executions (even if the algorithm answered Non-exhaustive).*

*You will precise the bound for which we can be certain the algorithm is complete.*

**Question 8** *However, in general, the minimal bound that ensures the algorithm is complete is lower that the upper bound you've just given.*

*Give that minimal bound depending on the properties of the paths that exist in the transition system and the configurations they visit (and explain why it is the minimal bound, i.e., if you call the algorithm on that bound minus 1, it might return Non-exhaustive).*

**Question 9** *Even better, we can use a SMT-solver to determine that bound.*

*Give an algorithm, in the spirit of the global algorithm, which by several calls to a SMT-solver (you will precise the formula) computes the lowest $k$ for which the Bounded Model-Checking algorithm is assured to be complete on a given Boolean system.*

*You will suppose that the you have vectors of variables $\vec{v}$ that represent a configuration of the program, and you have the predicate $I(\vec{v})$ that is true if $\vec{v}$ is an initial configuration, and $T(\vec{v_1}, \vec{v_2})$ that is true if there is a transition from $\vec{v_1}$ to $\vec{v_2}$. You also have the equality predicate over these vectors ($\vec{v_1} = \vec{v_2}$).*

*For simplicity (to keep a manageable formula), you will consider the system is complete, i.e., for every $\vec{v_1}$, there exists $\vec{v_2}$ such that $T(\vec{v_1}, \vec{v_2})$ is true.*

**Question 10** *Justify you actually only need a SAT-solver to express the above formula (only explain what part of the formula needs an encoding).*

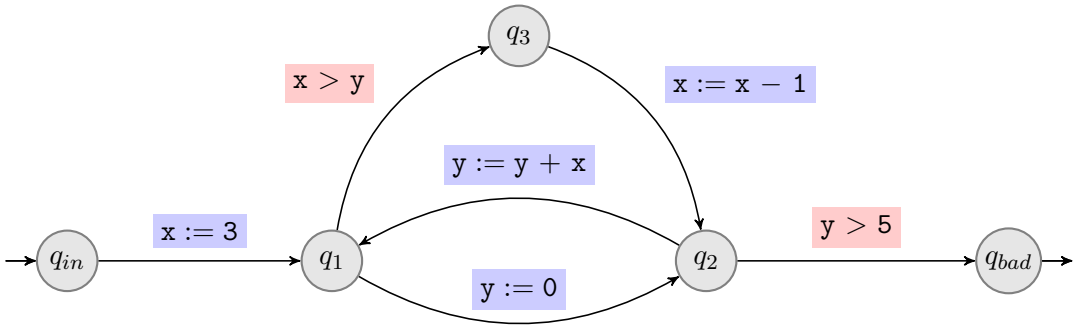
*That of course also applies to the formulæ of the Bounded Model-Checking algorithm. Did we just prove the Bounded Model-Checking over Boolean systems to be in NP? Why?*

# 2 Abstract Interpretation (10pts)

This section deals with abstract interpretation, and is divided into two independent subsections. The first subsection applies range analysis on an example. The second subsection shows that abstract interpretation, which was presented using Galois connections in the course, can also be phrased in terms of Moore families.

## 2.1 Range analysis

We perform range analysis — this analysis was presented in the course — on the control-flow automaton depicted below. This control-flow automaton has two variables `x` and `y`, both ranging over integers. The initial location is $q_{in}$ and the bad location is $q_{bad}$. Recall that range analysis uses the *abstract domain of intervals*.




Like in the course, an analysis will be called *successful* when the abstract value obtained for $q_{bad}$ is $\bot$. Round-robin iteration shall use the following order on locations: $q_{in}, q_1, q_2, q_3, q_{bad}$.

**Question 11** *Apply the round-robin algorithm with widening. Do* not *use narrowing. Is the analysis successful?*

**Question 12** *Starting from the result of the previous question, perform a decreasing iteration with narrowing. Is the analysis successful?*

## 2.2 Moore families

The main objective of this section is to show that abstract interpretation, which was presented using Galois connections in the course, can also be phrased in terms of so-called Moore families.

We start by recalling some notions and notations from the course. A *complete lattice* is a partially-ordered set $(L, \sqsubseteq)$ such that every subset $X \subseteq L$ admits a greatest lower bound $\sqcap X$ and a least upper bound $\sqcup X$. Recall that a complete lattice necessarily has a least element $\bot$ and a greatest element $\top$.

Given two complete lattices $(C, \sqsubseteq)$ and $(A, \preceq)$, a pair of functions $\alpha : C \to A$ and $\gamma : A \to C$ forms a *Galois connection*, written $(C, \sqsubseteq) \underset{\alpha}{\overset{\gamma}{\leftrightarrows}} (A, \preceq)$, when the equivalence $\alpha(c) \preceq a \Leftrightarrow c \sqsubseteq \gamma(a)$ holds for every $a \in A$ and $c \in C$.

We now introduce the notion of Moore family. Assume that we are given a complete lattice $(C, \sqsubseteq)$ with greatest lower bound written $\sqcap$ and least upper bound written $\sqcup$. A *Moore family* is a subset $M \subseteq C$ such that $\sqcap X \in M$ for every $X \subseteq M$. In words, a Moore family is a subset of $C$ that is closed under arbitrary greatest lower bounds.

**Question 13** *Prove that the greatest element $\top$ of $C$ is necessarily contained in $M$.*

**Question 14** *Is the least element $\bot$ of $C$ necessarily contained in $M$? Justify your answer.*

Let us consider the function $\alpha : C \to M$ defined by $\alpha(c) = \sqcap\{m \in M \mid c \sqsubseteq m\}$. The property that $\alpha(c) \in M$ is guaranteed by the assumption that $M$ is a Moore family. By definition, $\alpha(c)$ is a lower bound of the set $\{m \in M \mid c \sqsubseteq m\}$. The following question shows that $\alpha(c)$ is also in this set. Put differently, this means that $\alpha(c)$ is the mimimal over-approximation of $c$ by an element in $M$.

**Question 15** *Prove that for every $c \in C$, we have $c \sqsubseteq \alpha(c)$.*

**Question 16** *Prove that the partially-ordered set $(M, \sqsubseteq)$ is a complete lattice with greatest lower bound $\wedge$ and least upper bound $\vee$ verifying*

$$\wedge X \;=\; \sqcap X \qquad\qquad \vee X \;=\; \alpha(\sqcup X)$$

*for every subset $X \subseteq M$.*

**Question 17** *Prove that $(C, \sqsubseteq) \underset{\alpha}{\overset{id}{\leftrightarrows}} (M, \sqsubseteq)$ is a Galois connection.*[1]

We have shown that every Moore family induces a Galois connection. Let us now show the converse. Consider a Galois connection $(C, \sqsubseteq) \underset{\alpha}{\overset{\gamma}{\leftrightarrows}} (A, \preceq)$ between two complete lattices $(C, \sqsubseteq)$ and $(A, \preceq)$. The greatest lower bound and least upper bound are respectively denoted by $\sqcap$ and $\sqcup$ in $(C, \sqsubseteq)$, and by $\wedge$ and $\vee$ in $(A, \preceq)$. We first show that $\gamma$ preserves greatest lower bounds. This property was mentioned (without proof) in the course.

**Question 18** *Prove that for every subset $X \subseteq A$, the equality $\gamma(\wedge X) = \sqcap\{\gamma(x) \mid x \in X\}$ holds.*

**Question 19** *Deduce from the previous question that the set $M = \{\gamma(a) \mid a \in A\}$ is a Moore family.*

[1] The function $id : M \to C$ is the *identity* function (i.e., $id(x) = x$ for all $x \in M$).

## 3 Upward-closed Sets (8pts)

During the lecture, we presented the concept of upward-closed sets in order to manipulate some sets of natural numbers (or vectors of natural numbers). In this section we extend this concept from the natural numbers to the integers.

The totaly-ordered set of natural numbers $(\mathbb{N}, \geq)$ is extended with a bottom element $+\infty$ and a top element $-\infty$ (notice that $\geq$ is the reverse order of $\leq$ and thus $+\infty$ is really a bottom element and not a top one). This extended set is denoted by $(\bar{\mathbb{N}}, \geq)$. We associate with each element $b \in \bar{\mathbb{N}}$ the interval $\uparrow b$ defined as follows:

$$\uparrow b = \{z \in \mathbb{Z} \mid z \geq b\}$$

With such a definition, notice that $\uparrow +\infty = \emptyset$, $\uparrow 0 = \mathbb{N}$, and $\uparrow -\infty = \mathbb{Z}$.

We introduce the concretization function $\gamma : (\bar{\mathbb{N}}, \geq) \to (2^{\mathbb{Z}}, \subseteq)$ defined by $\gamma(b) = \uparrow b$ for every $b \in \bar{\mathbb{N}}$.

**Question 20** *Show that $\gamma$ is monotonic, i.e. satisfies $\gamma(a) \subseteq \gamma(b)$ if $a \geq b$ for every $a, b \in \bar{\mathbb{N}}$.*

**Question 21** *Provide the unique function $\alpha : (2^{\mathbb{Z}}, \subseteq) \to (\bar{\mathbb{N}}, \geq)$ such that $(\alpha, \gamma)$ forms a Galois connection, and prove that your function $\alpha$ satisfies the requirement.*

**Question 22** *Provide the value of $\alpha(\emptyset)$, $\alpha(\{2, 4\})$, $\alpha(\{0\})$ and $\alpha(\{-1\})$.*

**Question 23** *Prove that for every sequence $(a_n)_{n \in \mathbb{N}}$ of elements in $\bar{\mathbb{N}}$ there exists $i, j \in \mathbb{N}$ such that $i < j$ and $\gamma(a_j) \subseteq \gamma(a_i)$.*

Given a natural number $d \geq 1$ we denote by $\bar{\mathbb{N}}^d$ the set of $d$-dimensional vectors of elements in $\bar{\mathbb{N}}$. We introduce the partial-order $\geq$ over $\bar{\mathbb{N}}^d$ defined component-wise by $x \geq y$ if $x_i \geq y_i$ for every $i \in \{1, \ldots, d\}$. Given a vector $b \in \bar{\mathbb{N}}^d$, we denote by $\uparrow b$ the following subset of $\mathbb{Z}^d$:

$$\uparrow b = \uparrow b_1 \times \cdots \times \uparrow b_d$$

We also associate with a finite set $B \subseteq \bar{\mathbb{N}}^d$ the set $\uparrow B$ defined as $\bigcup_{b \in B} \uparrow b$.

**Question 24** *Provide an algorithm for deciding the inclusion $\uparrow A \subseteq \uparrow B$ for two finite sets $A, B \subseteq \bar{\mathbb{N}}^d$. Justify the correctness of your algorithm.*