Software Verification

Monday, January 11th 2021, 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 (10pts)

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

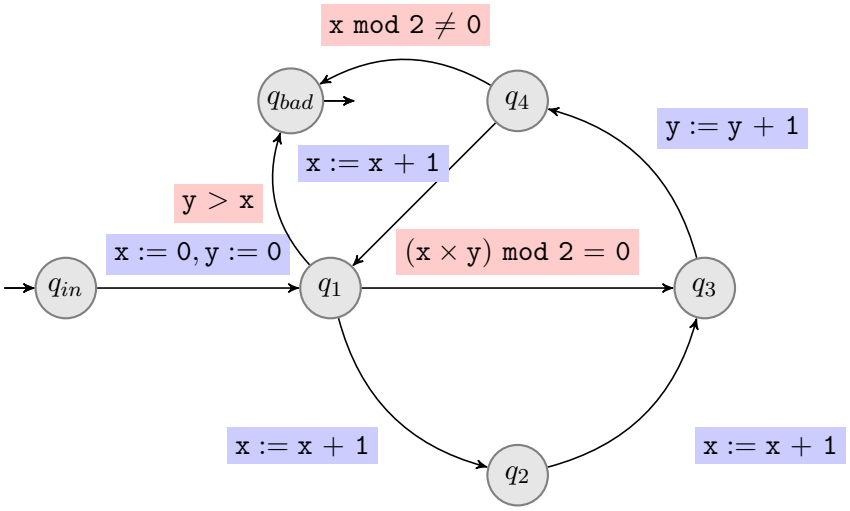
1.1 Principles

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 size 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.*



Question 3 *Apply the bounded model-checking algorithm to the previous system up to bound 7: 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.*

Is there a bound on which the algorithm would give a different answer? Justify.

1.2 Bounded Model-Checking with quantifier free FO-property

During the scope of the course, we focused on checking reachability properties, i.e., we actually only checked properties of the form "q_{bad} is not reachable". The objective of this part is to prove it is equivalent to check properties expressed in quantifier free First-Order Logic (FO).

Question 4 *We consider a CFA $\mathcal{A}$. In this question, we focus only on one of its states q_1. We consider the following formula $\varphi(q, x_1, \cdots, x_n) ::= (q = q_1) \Rightarrow (x_1 \leq 0)$.*

Give a CFA $\mathcal{A}[q_1, \varphi]$ (a drawing is perfectly acceptable) such that, if you replace the state q_1 of $\mathcal{A}$ with $\mathcal{A}[q_1, \varphi]$ (precise where in it you plug the input and output transitions of q_1), you get a new CFA $\mathcal{B}$ such that some state q_{bad} (of $\mathcal{A}[q_1, \varphi]$) is not reachable in $\mathcal{B}$ if and only if every reachable configuration $(q, x_1, \cdots, x_n)$ of $\mathcal{A}$ satisfies φ.

Question 5 *Prove the above equivalence.*

Question 6 *We now consider a more general formula $\varphi(q, x_1, \cdots, x_n) ::= (q = q_1) \Rightarrow \psi(\text{Var})$, where ψ is an arbitrary formula using only variables from* Var.

Inductively construct a CFA $\mathcal{A}[q_1, \varphi]$ as in question 4. To do so, do it by iterating on the structure of the formula (you have already done the base case in question 2, so you only need to explain the construction for conjunction and disjunction). You might need to name some subautomata.

Question 7 *Let us now consider the general case. We have a quantifier-free formula φ, that, without loss of generality, we suppose of the form $\bigwedge_{q_a \in Q_{\mathcal{A}}} ((q = q_a) \Rightarrow \psi_{q_a}(\text{Var}))$.*

Given a CFA $\mathcal{A}$, describe a CFA $\mathcal{A}_\varphi$ such that φ is satisfied on every reachable configuration of $\mathcal{A}$ if and only if a distinguished bad state is not reachable in $\mathcal{A}_\varphi$.

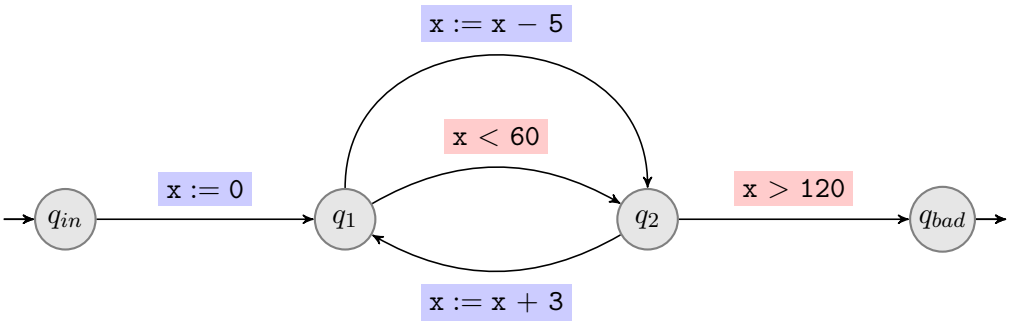
Question 8 *Suppose that, in the previous case, there is bug of length k in $\mathcal{A}$. With what bound (at worst) must you run a BMC algorithm on $\mathcal{A}_\varphi$ to find it?*

2 Abstract Interpretation (10pts)

This section deals with precision improvement in abstract interpretation based range analysis. The first subsection applies range analysis on a simple example, using the standard widening and narrowing operators (i.e., those presented in the course). The second subsection investigates a generic technique to refine widening operators and then applies this technique on the same simple example.

2.1 Range analysis with standard widening and narrowing

We perform range analysis — this analysis was presented in the course — on the control-flow automaton depicted below. This control-flow automaton has a single variable `x`, 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 is called *successful* when the abstract value obtained for q_{bad} is $\bot$. Round-robin iteration shall use the following order on locations: $q_{in}, q_1, q_2, q_{bad}$.

Question 9 *Apply the round-robin algorithm with widening. Do* not *use narrowing. Is the analysis successful?*

Question 10 *Starting from the result of the previous question, perform a decreasing iteration with narrowing. Is the analysis successful?*

2.2 Widening up-to and application to range analysis

As observed in the course and in the previous subsection, a widening operator may introduce too much imprecision. We shall remedy this situation using a simple technique called widening "up-to".

Let us first recall some notions and notations from the course. Consider a complete lattice $(L, \sqsubseteq)$, with greatest lower bound written $\sqcap$ and least upper bound written $\sqcup$. A *widening* operator for $(L, \sqsubseteq)$ is a function $\nabla : (L \times L) \to L$ such that:

1. for every $x, y \in L$, it holds that $(x \sqcup y) \sqsubseteq (x \nabla y)$, and

2. for every ascending chain $x_0 \sqsubseteq x_1 \sqsubseteq \cdots$ of elements of L, the ascending chain $y_0 \sqsubseteq y_1 \sqsubseteq \cdots$ defined by

$$\begin{cases} y_0 &= x_0 \\ y_{i+1} &= y_i \nabla x_{i+1} \quad \text{for all } i \in \mathbb{N} \end{cases}$$

is not strictly increasing (i.e., $y_{i+1} = y_i$ for some $i \in \mathbb{N}$).

The remainder of this section focuses on a generic technique to refine widening operators. Assume that we are given firstly a widening operator ∇ for a complete lattice $(L, \sqsubseteq)$, and secondly a finite subset M of L. We let ∇_M^{upto} denote the function $\nabla_M^{\text{upto}} : (L \times L) \to L$ defined by

$$x \nabla_M^{\text{upto}} y \;=\; (x \nabla y) \sqcap \bigsqcap \{m \in M \mid x \sqsubseteq m \wedge y \sqsubseteq m\} \;.$$

The idea behind this definition is to apply the widening operator ∇ and then refine its result by "intersecting" it with the elements of M that are greater than or equal to both arguments x and y. The first observation is that ∇_M^{upto} is always more precise than ∇.

Question 11 *Prove that for every $x, y \in L$, it holds that $(x \nabla_M^{\text{upto}} y) \;\sqsubseteq\; (x \nabla y)$.*

The second observation is that ∇_M^{upto} verifies the first condition[1] in the definition of widening operators.

Question 12 *Prove that for every $x, y \in L$, it holds that $(x \sqcup y) \;\sqsubseteq\; (x \nabla_M^{\text{upto}} y)$.*

We now restrict our attention to range analysis. This means that the complete lattice $(L, \sqsubseteq)$ is assumed to be the abstract lattice of intervals $(\mathsf{Int}, \sqsubseteq)$, and the widening operator ∇ is assumed to be the standard widening operator over intervals (i.e., the one that was used in the previous subsection). Both have been presented in the course. The following fact, which is admitted, allows us to replace ∇ by ∇_M^{upto} in the round-robin algorithm.

Fact 1 (Admitted) *For every finite subset $M \subseteq \mathsf{Int}$, the function ∇_M^{upto} is a widening operator for $(\mathsf{Int}, \sqsubseteq)$.*

For the two following questions, we consider the finite set $M \subseteq \mathsf{Int}$ defined by

$$M \;=\; \{(-15, 7), (-7, +\infty), (-\infty, 78), (-\infty, 82)\} \;.$$

Question 13 *Compute the following abstract intervals and write the details of your computations:*

1. $\bot \nabla_M^{\text{upto}} (0, 0)$,
2. $(0, 0) \nabla_M^{\text{upto}} (-2, 3)$,
3. $(-10, 0) \nabla_M^{\text{upto}} (-10, 5)$,
4. $(-\infty, 99) \nabla_M^{\text{upto}} (-\infty, 100)$,

Question 14 *Apply the round-robin algorithm with widening ∇_M^{upto} and* without *narrowing on the control-flow automaton depicted in Subsection 2.1. Is the analysis successful?*

[1] In fact, ∇_M^{upto} is not a widening operator in general as the second condition is not necessarily satisfied.

3 Upward-closed Sets (10pts)

Given a natural number d, we denote by $\mathbb{N}^d$ the set of d-dimensional vectors over the set of natural numbers $\mathbb{N}$. We extend the total order $\leq$ over the natural numbers $\mathbb{N}$ as a partial order, also denoted as $\leq$, over $\mathbb{N}^d$ defined by $\vec{x} \leq \vec{y}$ if there exists $\vec{v} \in \mathbb{N}^d$ such that $\vec{y} = \vec{x} + \vec{v}$ where the sum is defined component-wise.

Given a vector $\vec{x} \in \mathbb{N}^d$, we denote by $\uparrow \vec{x}$ the set $\{\vec{v} \in \mathbb{N}^d \mid \vec{x} \leq \vec{v}\}$. Given a set $\vec{X} \subseteq \mathbb{N}^d$, we also introduce the set $\uparrow \vec{X}$ defined as $\bigcup_{\vec{x} \in \vec{X}} \uparrow \vec{x}$. The set $\vec{X}$ is said to be *upward-closed* if $\vec{X} = \uparrow \vec{X}$.

Question 15 *Assume that $\vec{X}_1$ and $\vec{X}_2$ are two upward-closed sets. Show that $\vec{X}_1 \cap \vec{X}_2$ and $\vec{X}_1 \cup \vec{X}_2$ are upward-closed.*

A *basis* of an upward-closed set $\vec{U}$ is a set $\vec{B} \subseteq \vec{U}$ such that $\vec{U} = \uparrow \vec{B}$.

3.1 Algorithms for Inclusion, Union, and Intersection

In this section, we assume that $\vec{B}_1$ and $\vec{B}_2$ are two finite bases of two upward-closed sets $\vec{X}_1$ and $\vec{X}_2$.

Question 16 *Provide an algorithm deciding $\vec{X}_1 \subseteq \vec{X}_2$ from $\vec{B}_1$ and $\vec{B}_2$.*

Question 17 *Provide an algorithm computing a finite basis of $\vec{X}_1 \cup \vec{X}_2$ from $\vec{B}_1$ and $\vec{B}_2$.*

Given two vectors $\vec{b}_1$ and $\vec{b}_2$ in $\mathbb{N}^d$. We denote by $\max(\vec{b}_1, \vec{b}_2)$ the vector $\vec{b}$ in $\mathbb{N}^d$ defined by $\vec{b}(i) = \max(\vec{b}_1(i), \vec{b}_2(i))$ for every $1 \leq i \leq d$.

Question 18 *Show that $(\uparrow \vec{b}_1) \cap (\uparrow \vec{b}_2) = \uparrow \max(\vec{b}_1, \vec{b}_2)$.*

Question 19 *Provide an algorithm computing a finite basis of $\vec{X}_1 \cap \vec{X}_2$ from $\vec{B}_1$ and $\vec{B}_2$.*

3.2 Characteristic Bases

A vector $\vec{m}$ in a set $\vec{X} \subseteq \mathbb{N}^d$ is said to be *minimal* if for every $\vec{x} \in \vec{X}$ the relation $\vec{x} \leq \vec{m}$ implies $\vec{x} = \vec{m}$. The set of minimal elements of $\vec{X}$ is denoted by $\min \vec{X}$. Let us recall from the Dickson's Lemma that $\min \vec{U}$ is a finite basis of $\vec{U}$ for every upward-closed set $\vec{U}$. The set $\min \vec{U}$ is called the *characteristic basis* of $\vec{U}$.

Question 20 *Assume that $\vec{B}_1$ and $\vec{B}_2$ are the characteristic bases of two upward-closed sets $\vec{U}_1$ and $\vec{U}_2$. Show that the equality $\vec{U}_1 = \vec{U}_2$ is equivalent to the equality $\vec{B}_1 = \vec{B}_2$.*

Question 21 *Assume that $\uparrow \vec{X} = \uparrow \vec{Y}$ for two sets $\vec{X}, \vec{Y} \subseteq \mathbb{N}^d$. Show that $\min \vec{X} \subseteq \min \vec{Y}$. Deduce that $\min \vec{X} = \min \vec{Y}$ in that case.*

Question 22 *Assume that $\vec{B}$ is a basis of an upward-closed set $\vec{U}$. Show that $\min \vec{B}$ is the characteristic basis of $\vec{U}$.*