Software Verification

Wednesday, January 9th 2019, 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 predicate abstraction.

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.*

1.2 Wolf, Goat and Cabbage

The goal of this exercise is to model the following problem and use the Bounded-Model Checking to solve it. Of course, here, the "counter-example" will represent the desired output and not a faulty execution.

> "*A farmer travels with a cabbage, a goat and a wolf. He wants to cross a river but there is only a small boat which can only carry himself plus one of its animal or vegetable (the cabbage is very big). The problem is, if he leaves the goat with the cabbage, or the wolf with the goat without him being present, there will be casualties. How can the river can be crossed without anyone being eaten?*"

Question 3 *Give an automaton representing the problem (it should contain 16 states if you represent everything), and mark the goal state (everybody is at the other side of the river). Use clear and short names for the states. Don't put outgoing transitions on the states where someone gets eaten. For the sake of simplicity, you can merge all the states where someone gets eaten.*

Question 4 *Perform the Bounded Model-Checking with the Depth-First Search algorithm, with the exploration order nobody $<$ cabbage $<$ goat $<$ wolf and max depth 10. You should represent the algorithm as a tree (you can stop at the first solution found).*

Question 5 *Is there a shorter solution? If so, give it. How could you prove that a solution is the shortest one?*

2 Abstract Interpretation with Closure Operators (10pts)

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 closure operators.

We start by recalling some basic notions. A function $f : L \to L$ on a complete lattice $(L, \sqsubseteq)$ is called

- *monotonic* if $(x \sqsubseteq y \implies f(x) \sqsubseteq f(y))$ for every $x, y \in L$,
- *reductive* if $f(x) \sqsubseteq x$ for every $x \in L$,
- *extensive* if $f(x) \sqsupseteq x$ for every $x \in L$,
- *idempotent* if $f(x) = f(f(x))$ for every $x \in L$.

We also recall the characterization of Galois connections that was given in the course. 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 $(C, \sqsubseteq) \underset{\alpha}{\overset{\gamma}{\leftrightarrows}} (A, \preceq)$, if, and only if, α and γ are monotonic, $\alpha \circ \gamma$ is reductive, and $\gamma \circ \alpha$ is extensive.

We now introduce the notion of closure operator and show that every Galois connection induces a closure operator. A *closure operator* on a complete lattice $(L, \sqsubseteq)$ is a function $cl : L \to L$ that is monotonic, extensive and idempotent.

Question 6 *Prove that, for every Galois connection $(C, \sqsubseteq) \underset{\alpha}{\overset{\gamma}{\leftrightarrows}} (A, \preceq)$, the function $\gamma \circ \alpha$ is a closure operator.*

The proof that every closure operator induces a Galois connection will require more work. For the remainder of this section, we fix a complete lattice $(L, \sqsubseteq)$ and a closure operator cl on L. The greatest lower bound and least upper bound of $(L, \sqsubseteq)$ are written $\sqcap$ and $\sqcup$, respectively.

An element $x \in L$ is called *closed* if $cl(x) = x$. Observe that, by idempotence, the set $cl(L) \stackrel{\text{def}}{=} \{cl(x) \mid x \in L\}$ is the set of all closed elements of L. The following lemma is admitted without proof, and can be used to solve the next question.

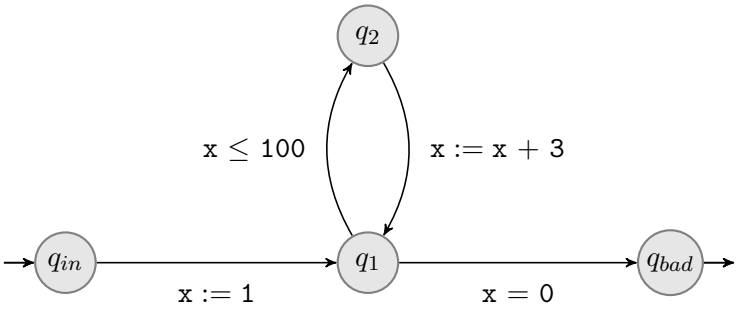

Figure 1: Control-flow automaton of Question 9.

Lemma 1 *Let M be a subset of L. If $\sqcap X$ is in M for every subset $X \subseteq M$, then $(M, \sqsubseteq)$ is a complete lattice.*

Question 7 *Prove that that $(cl(L), \sqsubseteq)$ is a complete lattice.*

Question 8 *Prove that $(L, \sqsubseteq) \underset{cl}{\overset{id}{\leftrightarrows}} (cl(L), \sqsubseteq)$ is a Galois connection.*[1]

We conclude this section with an application to program analysis by abstract interpretation. Consider the function $\theta : \mathcal{P}(\mathbb{Z}) \to \mathcal{P}(\mathbb{Z})$ defined by

$$\theta(X) = \begin{cases} X & \text{if X is finite and } |X| \leq 3 \\ \mathbb{Z} & \text{otherwise} \end{cases}$$

In the above definition, $|X|$ denotes the cardinal of X. It is easily seen that θ is a closure operator on the complete lattice $(\mathcal{P}(\mathbb{Z}), \subseteq)$. Therefore, $(\theta(\mathcal{P}(\mathbb{Z})), \subseteq)$ is a complete lattice by Question 7 and $(\mathcal{P}(\mathbb{Z}), \subseteq) \underset{\theta}{\overset{id}{\leftrightarrows}} (\theta(\mathcal{P}(\mathbb{Z})), \subseteq)$ is a Galois connection by Question 8.

Consider the control-flow automaton depicted in Figure 1. This control-flow automaton has one variable `x`, which ranges over integers. The initial location is q_{in} and the bad location is q_{bad}. Recall that '`:=`' denotes an assignment and that '`=`' denotes a condition. Like in the course, an analysis will be called *successful* when the abstract value obtained for q_{bad} has an empty concretization. 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 to analyze the control-flow automaton depicted in Figure 1 using the Galois connection $(\mathcal{P}(\mathbb{Z}), \subseteq) \underset{\theta}{\overset{id}{\leftrightarrows}} (\theta(\mathcal{P}(\mathbb{Z})), \subseteq)$. Is the analysis successful?*

3 Predicate Abstraction (10pts)

In the sequel, X is a finite set of *variables* of a program, and a *valuation* is a mapping $\rho : X \to \mathbb{Z}$. A predicate p is a formula over X. We denote by $[\![p]\!]$ the set of valuations

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

satisfying p. We denote by $\mathbb{T}$ the four-values domain $\{\top, \bot, \text{false}, \text{true}\}$ equipped with the partial order $\sqsubseteq$ defined by $s \sqsubseteq t$ if $s = \bot \vee t = \top \vee s = t$.

Let P be a finite set of predicates. We introduce the set of trivectors $P \to \mathbb{T}$ partially ordered by the component-wise extension $\sqsubseteq_P$ of $\sqsubseteq$ defined by $s \sqsubseteq_P t$ if $s(p) \sqsubseteq t(p)$ for every $p \in P$. We also introduce the Galois connection $(\mathcal{P}(X \to \mathbb{Z}), \subseteq) \underset{\alpha_P}{\overset{\gamma_P}{\leftrightarrows}} (\mathbb{T}^P, \sqsubseteq_P)$ where α_P is defined for every set V of valuations and for every predicate p in P by:

$$\alpha_P(V)(p) = \begin{cases} \bot & \text{if } V = \emptyset \\ \text{true} & \text{if } \emptyset \neq V \subseteq [\![p]\!] \\ \text{false} & \text{if } \emptyset \neq V \subseteq [\![\neg p]\!] \\ \top & \text{otherwise} \end{cases}$$

Question 10 *Explain the following inclusion:*

$$\{\alpha_P(V) \mid V \in \mathcal{P}(X \to \mathbb{Z})\} \subseteq \{P \to \{\bot\}\} \cup \{P \to \{\text{true}, \text{false}, \top\}\}$$

Question 11 *Provide a formula over X that is unsatisfiable if, and only if, $\gamma_P(t)$ is empty where $t : P \to \mathbb{T}$ is a trivector. An answer without a proof will not be considered.*

We introduce a set X' of disjoint copies of X. The copy of a variable x in X is the variable x' in X'. Given a predicate p, we denote by p' the formula obtained from p by replacing every variable x by its copy x'.

We denote by Op the set of *operations* over X. Given an operation op in Op, we denote by $\xrightarrow{op}$ the binary relation over the valuations defined by $\rho \xrightarrow{op} \rho'$ if ρ' is the valuation obtained from ρ after executing op. We also denote by ϕ_{op} the formula over $X \cup X'$ such that a mapping $\mu : X \cup X' \to \mathbb{Z}$ is satisfying ϕ_{op} if, and only if, $\rho \xrightarrow{op} \rho'$ where ρ, ρ' are the valuations defined by $\rho(x) = \mu(x)$ and $\rho'(x) = \mu(x')$ for every $x \in X$.

We are interested in analyzing programs with trivectors. During the lecture, we associated a finite set of predicates to a program. In order to improve the scalability of the analysis, we are interested in associating a finite set of predicates to each location of the program. In order to perform an analysis, we must provide the effect in the trivector abstract domain of a transition labeled by an operation op from a location with a set P of predicates to a location with a set Q of predicates.

To do so, let $s : P \to \mathbb{T}$ be a trivector and let us introduce the trivector $t : Q \to \mathbb{T}$ defined as follows:

$$t = \alpha_Q(\{\rho' : X \to \mathbb{Z} \mid \exists \rho \in \gamma_P(s) \mid \rho \xrightarrow{op} \rho'\})$$

Let q be a predicate in Q.

Question 12 *Provide a formula over $X \cup X'$ that is unsatisfiable if, and only if $t(q) \in \{\bot, \text{true}\}$. An answer without a proof will not be considered.*

Question 13 *Provide a formula over $X \cup X'$ that is unsatisfiable if, and only if $t(q) \in \{\bot, \text{false}\}$. An answer without a proof will not be considered.*

Question 14 *Deduce an algorithm for computing t.*

Question 15 *Explain how to modify the CEGAR loop in order to perform an analysis such that the sets of predicates depend on the locations.*

Question 16 *Explain why the new algorithm is progressing at each iteration of the CEGAR loop.*