CS402 End-Semester Examination

Max marks: 65 **Time: 3 hours**

- *Be brief, complete and stick to what has been asked.*
- *Unless asked for explicitly, you may cite results/proofs covered in class without reproducing them.*
- ***If you need to make any assumptions, state them clearly.***
- ***Do not copy solutions from others. Penalty for offenders: FR grade.***

1. In class, we have discussed how the imprecision introduced by the *lub* operation in an abstract domain A can be refined by using a different abstract domain whose elements are appropriate subsets of the elements of A. In this question, we will try to make this idea a bit more concrete.

   Let $L = (A, \sqsubseteq, \sqcup, \sqcap, \top, \bot)$ be a (topped and bottomed) lattice with the underlying set A. Define A^p to be $\{S \mid S \subseteq A, \forall x, y \in S,\ x \not\sqsubseteq y \text{ and } y \not\sqsubseteq x\}$. In other words, A^p is the set of all subsets of A such that no two elements in the subset are ordered by $\sqsubseteq$. Define the binary relation $\sqsubseteq^p \subseteq A^p \times A^p$ as follows: $\forall S_1, S_2 \in A^p.\ S_1 \sqsubseteq^p S_2$ if and only if $\forall x \in S_1\ \exists y \in S_2.\ x \sqsubseteq y$. Similarly, define $\top^p = \{\top\}$ and $\bot^p = \{\bot\}$. We now wish to define two binary functions $\sqcup^p : A^p \times A^p \to A^p$ and $\sqcap^p : A^p \times A^p \to A^p$ such that $L^p = (A^p, \sqsubseteq^p, \sqcup^p, \sqcap^p, \top^p, \bot^p)$ forms a lattice. In other words, we want $\sqcup^p$ and $\sqcap^p$ to act as the *lub* and *glb* operators in L^p.

   - *[5 marks]* Show that $\sqsubseteq^p$ as defined above is an anti-symmetric relation.
   - *[5 marks]* Show that there exists a binary function $\sqcup^p$ that computes the *least upper bound* (in the ordering $\sqsubseteq^p$) of every pair of elements in A^p? You must give an unambiguous definition of $\sqcup^p$.
   - *[5 marks]* Show that there exists a binary function $\sqcap^p$ that computes the *greatest lower bound* (in the ordering $\sqsubseteq^p$) of every pair of elements in A^p? You must give an unambiguous definition of $\sqcap^p$.
   - *[10 marks]* Suppose $\nabla : L \times L \to L$ is a widening operator in L. Can you define a widening operator in L^p using ∇, $\sqcap^p$, $\sqcup^p$ and $\subseteq^p$? If yes, you must give an unambiguous definition of ∇^p and argue why it satisfies all the properties required of a widening operator.

2. Consider the following program in the language studied in class. Here `ptrType` denotes a pointer type to a structure that itself has a single field named `n` of type `ptrType`.

```
  function foo(ptrType y, ptrType z)

  // assume y and z point to (possibly shared) nil-terminated lists
  // and z is not nil

   ptrType x = z;
L1:  while ((x != y) and (x != nil)) do
L2:     x := x->n;
L3:  // end of while loop
L4:  return;
```

(a) *[5 marks]* Under the assumption that `y` and `z` point to (possibly shared) `nil`-terminated lists, what is the maximum number of heap-shared nodes that we can have at any time during the execution of the function ? You must provide justification for your answer.

(b) *[15 marks]* Suppose your solution to the above sub-question is n. We will use `A1, ..., An` as names of heap-shared nodes, and assume the following ordering of names: $x \prec y \prec z \prec A1 \prec \ldots \prec An$.

Clearly, when the function terminates, either `x = nil` or `x == y`. We wish to determine which of these possibilities is the right one given the initial state of the heap when function `foo` is called. Towards this end, we wish to develop a *custom* automata-based technique as follows.

- We will encode the heap using a word as in regular model checking. Specifically, this word must have a segment encoding each uninterrupted list in the heap.
- We will design a regular transducer that takes the encoding of the heap prior to executing the statement at `L2` and transforms it to the encoding of the heap obtained immediately after execution of `L2`. Note that we aren't asking for the transducer to change the encoding of the state (which includes program counter values, list of `nil`-valued variables, list of uninitialized variables, mode flags, etc.)
- We will then check whether the encoding of the heap obtained by repeated transformations of the statement at `L2` either contains the substring xy or does not contain x. In the former case, and under the assumptions stated in the program, the program terminates with `x == y`; otherwise, it terminates with `x = nil`.

Design the regular transducer referred to in the second step of the above outline. You may label edges of the transducer with s_1/s_2, where s_1, s_2 are strings (of zero or more letters).

(c) *[2.5 + 2.5 marks]* Suppose the transduction function defined by your transducer is R. Consider the two heaps encoded by $\sigma_1 = xz.n.\bot \mid y.n.\bot$ and $\sigma_2 = xz.n.yA1 \mid yA1.n.\bot$. Give the strings $R(\sigma_1)$ and $R(\sigma_2)$ obtained using your transducer in each case.

3. *[15 marks]* Consider the program given below, where all variables are of `int` type.

```
  function foo(int y, int z)

  int x;

L1:   x := z;
L2:  while ((x != y) and (x != 0)) do
L3:     x := x - 1;
L4:  return(x);
```

We wish to determine if $x = y$ or $x = 0$ when the function returns.

Using at most 5 predicates, design a Boolean program that allows you to answer the above question as precisely as possible. You are free to choose any 5 predicates on x, y and z. However, you must justify your choice of predicates, and give the corresponding Boolean program.