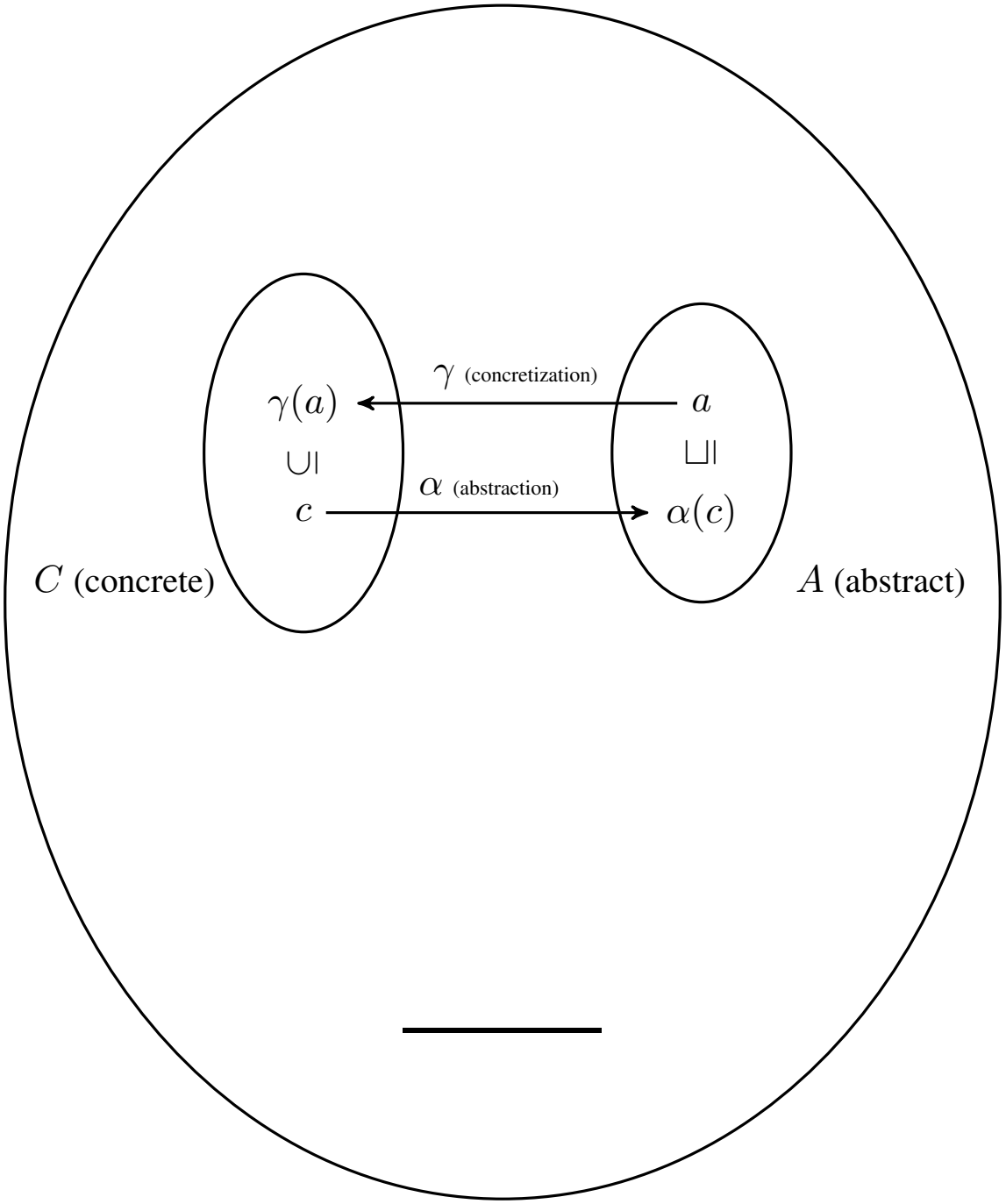
Quiz

Synthesis, Analysis, and Verification 2013

Friday, May 3rd, 2013



Problem 1: Recursion ([20 points])

Assume x, y and r are the only global variables in the program, and that their values range over the integers $\mathbb{Z}$. Consider the following function:

```
def f: Unit = {
  if(y < 0) {
      y = y + 1
      f
      r = r − x
  } else {
    if (y != 0) {
      y = y − 1
      f
      r = r + x
    }
  }
}
```

Task a) Derive the transformer $E(r_f)$ for this function, where r_f is the relation corresponding to the function. Give the signature ("type declaration") of this mathematical function E.

Task b) Define the relation $\sigma \subseteq \mathbb{Z}^3 \times \mathbb{Z}^3$ by

$$\sigma = \{((x, y, r), (x', y', r')) \mid r' = r + x * y\}$$

For this σ and for E from task a), is it the case that $E(\sigma) \subseteq \sigma$? Prove or give a counterexample.

Task c) Does the function satisfy the following specification?

$$r' = r + x * y$$

Show or disprove using and, if necessary, adapting your answer from tasks a) and b).

Problem 2: Transitive closure ([20 points])

Let A be a set and let $\Delta = \{(a, a) \mid a \in A\}$ be the identity relation. Recall that for each $r, r_1, r_2 \subseteq A \times A$ and each $P \subseteq A$

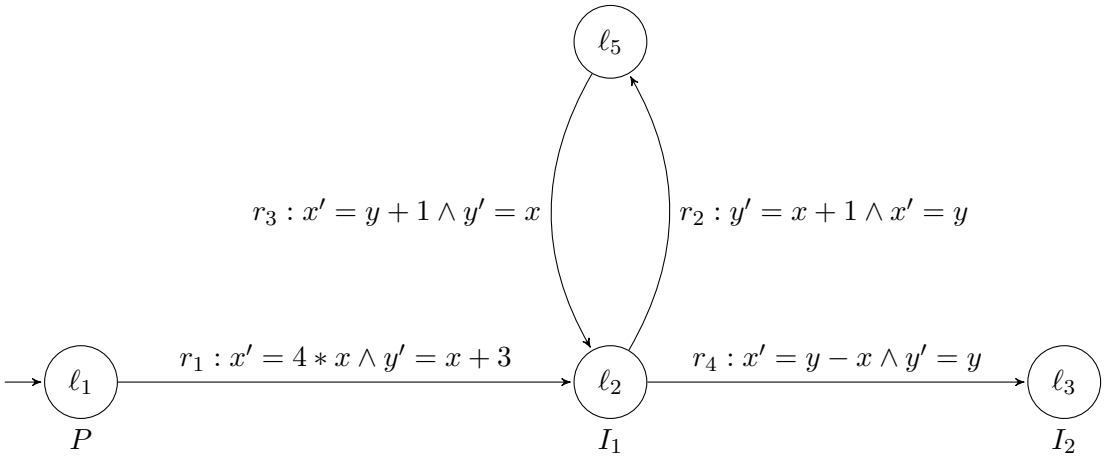
$$\begin{array}{rcl} sp(P, r) & = & \{s' \mid \exists s \,.\, s \in P \wedge (s, s') \in r\} \\ r_1 \circ r_2 & = & \{(s, s') \mid \exists t \,.\, (s, t) \in r_1 \wedge (t, s') \in r_2\} \\ r^0 & = & \Delta \\ r^i & = & r^{i-1} \circ r \qquad (\text{if } i > 0) \\ r^* & = & \bigcup_{i=0}^{\infty} r^i \\ r^+ & = & \bigcup_{i=1}^{\infty} r^i \end{array}$$

Easy consequences thus are: $r^+ \subseteq r^*$ and that $r^+ = r^* \circ r$.

In the rest of this question we identify a formula P with free variables x, y and the set $\{(x, y) \mid P\}$. Similarly, we make no difference between the formula r with free variables x, y, x', y' and the corresponding relation $\{((x, y), (x', y')) \mid r\}$.

Task a) Let $S = sp(P, r^*)$. Prove that $sp(S, r) \subseteq S$.

Task b) Consider a program whose control flow graph is shown in the figure below.



The program's precondition is $P = x \geq 0 \wedge y \leq -5$. Compute the invariants I_1, I_2 at control locations ℓ_2, ℓ_3. Your invariants should be in Presburger arithmetic, and be the strongest, i.e. the most precise with respect to the precondition P. Try to find the simplest form of the invariants, in particular, eliminate all quantifiers where possible.

Task c) The relations on all the transitions are Presburger arithmetic relations. State a syntactic constraint on r_2 and r_3 that made it possible to express the strongest invariants I_1, I_2 from part b) in quantifier-free Presburger arithmetic.

Problem 3: Hoare logic ([20 points])

Suppose we have a language that supports integers, booleans, sets and lists.
A set will be modeled as a regular mathematical set. We will model a list as a relation L such that $(i, v) \in L$ means that the element v is at position i in the list. We define

$$size(L) = \max\{i \mid (i, v) \in L\}$$

Indices should be non-negative and no two different elements should share the same index.

Task a) Give the condition that a relation L must satisfy to represent a list according to our description above.

For the next parts, consider the following program, where **:+** is the operator that adds an element at the end of a list, so e.g. List(1,2,3) :+ 4 = List(1,2,3,4). Assume that the assignment on line 3 does a deep copy, or, equivalently, that all lists are immutable mathematical values whose internal structure does not change.

```
1  // Precondition : ∀v.v ∈ S0 → v ≥ 0
2  def weirdSort(S0: Set[Int]) {
3    var k = 0
4    var S = S0
5    var L: List[Int] = List.empty
6
7    // Condition:
8    while // Invariant:
9            (!S.isEmpty) {
10     if (S.contains(k)) {
11       L = L :+ k
12       S = S − k
13     }
14     k = k + 1
15   }
16 }
17 // Postcondition :
18 ∀i, j, v, w.(i, v) ∈ L ∧ (j, w) ∈ L ∧ i < j → v < w
19 ∀v ∈ S0.∃i.(i, v) ∈ L
```

Task b) Give the strongest condition that is satisfied just before the loop and after the initialisation.

Task c) Find an appropriate loop invariant, and use it to prove that, whenever we run the above program in a state that satisfies the Precondition, its final state satisfies the Postcondition. You need to explain

1. why the invariant holds initially in all states that satisfy the precondition
2. why it is inductive (preserved on each execution of the loop body starting from any state satisfying only the invariant)
3. why it can be used to prove the Postcondition

State each of these conditions as a Hoare triple, and prove it. You will need to translate the list append operator to our model on lists. Your proof of individual Hoare triples need not be very detailed. Feel free to use any notation of sets, relations, and quantifiers in your invariants and Hoare triples.

Problem 4: Galois connection ([20 points])

Recall that a Galois connection is defined by two monotonic functions $\alpha : C \to A$ and $\gamma : A \to C$ between partial orders $(C, \subseteq)$ and $(A, \sqsubseteq)$ such that

$$\forall c \in C.\ \forall a \in A. \qquad c \subseteq \gamma(a) \ \Leftrightarrow\ \alpha(c) \sqsubseteq a$$

Let S and T be sets and r be a relation

$$r \subseteq S \times T$$

Think of $(s, t) \in r$ as saying that s and t are compatible.
Define functions between $(2^S, \subseteq)$ and $(2^T, \supseteq)$ like this:

$$\alpha(P) = \{t \in T \mid \forall s \in P.(s, t) \in r\}$$
$$\gamma(Q) = \{s \in S \mid \forall t \in Q.(s, t) \in r\}$$

Task a) Is (α, γ) a Galois connection? Prove or disprove.

Task b) Is any of α, γ guaranteed to be surjective or injective? Prove or give a counterexample.

Task c) Let T be a set of formulas over some set of variables Var of type U. Let S be the set of all functions $\mathsf{Var} \to U$, mapping variable to values. Define $(s, t) \in r$ to hold if the formula t is true for the values of variables given by s. Do you obtain a Galois connection for such r? Which abstract interpretation mentioned in the course does this correspond to?

Task d) Let S and T be sets of states and let z be a command whose meaning is $\rho(z) \subseteq S \times T$. Define r as the complement $\overline{\rho(z)}$. Define α and γ for such compatibility relation r. Express γ using wp and α using sp. What does then the Galois connection condition for such α and γ tell us about the relationship between sp and wp?

Problem 5: Widening ([20 points])

Given $\alpha : (C, \subseteq) \to (A, \sqsubseteq)$ and $\gamma : (A, \sqsubseteq) \to (C, \subseteq)$, suppose that (α, γ) is a Galois connection.

Task a) Define $\alpha'(c) = \gamma(\alpha(c))$ for all $c \in C$. Show that:

1. α' is monotonic;
2. the result of α' is larger or equal than its argument;
3. $\alpha'(\alpha'(c)) = \alpha'(c)$ for every c.

Task b) Let $C = 2^{\mathcal{Z}}$. Consider the following endpoints of intervals:

$$E = \{-\infty, -1000, -100, 0, 100, 1000, \infty\}$$

Define $A \subseteq E \times E$ by

$$A = \{(p, q) \mid p, q \in E, p \leq q\} \cup \{\bot\}$$

where we assume $-\infty \leq x$ and $x \leq \infty$ for every x. Define $\gamma : A \to C$ by

$$\gamma(a) = \begin{cases} \{x \mid p \leq x \leq q\} & \text{if } a = (p, q) \\ \emptyset & \text{if } \bot \end{cases}$$

and $\alpha : C \to A$ by

$$\alpha(c) = \begin{cases} (p, q) & \text{if } c \neq \emptyset \\ \bot & \text{if } c = \emptyset \end{cases}$$

where, for $c \in C$

$$p = \max\{e \in E \mid \forall x \in c.\ e \leq x\}$$
$$q = \min\{e \in E \mid \forall x \in c.\ x \leq e\}$$

Write the type signature of α' from part a) for this particular (α, γ). Write the definition of this α' and apply it to an example argument set of integers.

Task c) For the specific set E from task b) and $n \geq 0$, compute a constant H with the following property. Suppose we have a program with one integer variable whose control-flow graph has n points. Let F be the precise semantic transformer function $C^n \to C^n$ for this program, mapping current approximation of reachable states at each program point into a better approximation of reachable states. Then H should be such that $F(I) \bar{\subseteq} I$, for

$$I = (F')^H(\bar{\emptyset})$$

where $F'(x) = \bar{\alpha}'(F(x))$ for all x, and $\bar{\alpha}'(c_1, \ldots, c_n) = (\alpha'(c_1), \ldots, \alpha'(c_n))$ and is defined on the product lattice $(C^n, \bar{\subseteq})$.

Task d) Consider the following program

```
1    x = 0
2    while(x < 1000) {
3      x = x + 3
4    }
```

Let $F : C^n \to C^n$ be the transformer function for this program. Compute

$$I = (F')^H(\bar{\emptyset})$$

where H is as computed in part c). Compute also $F(I)$.