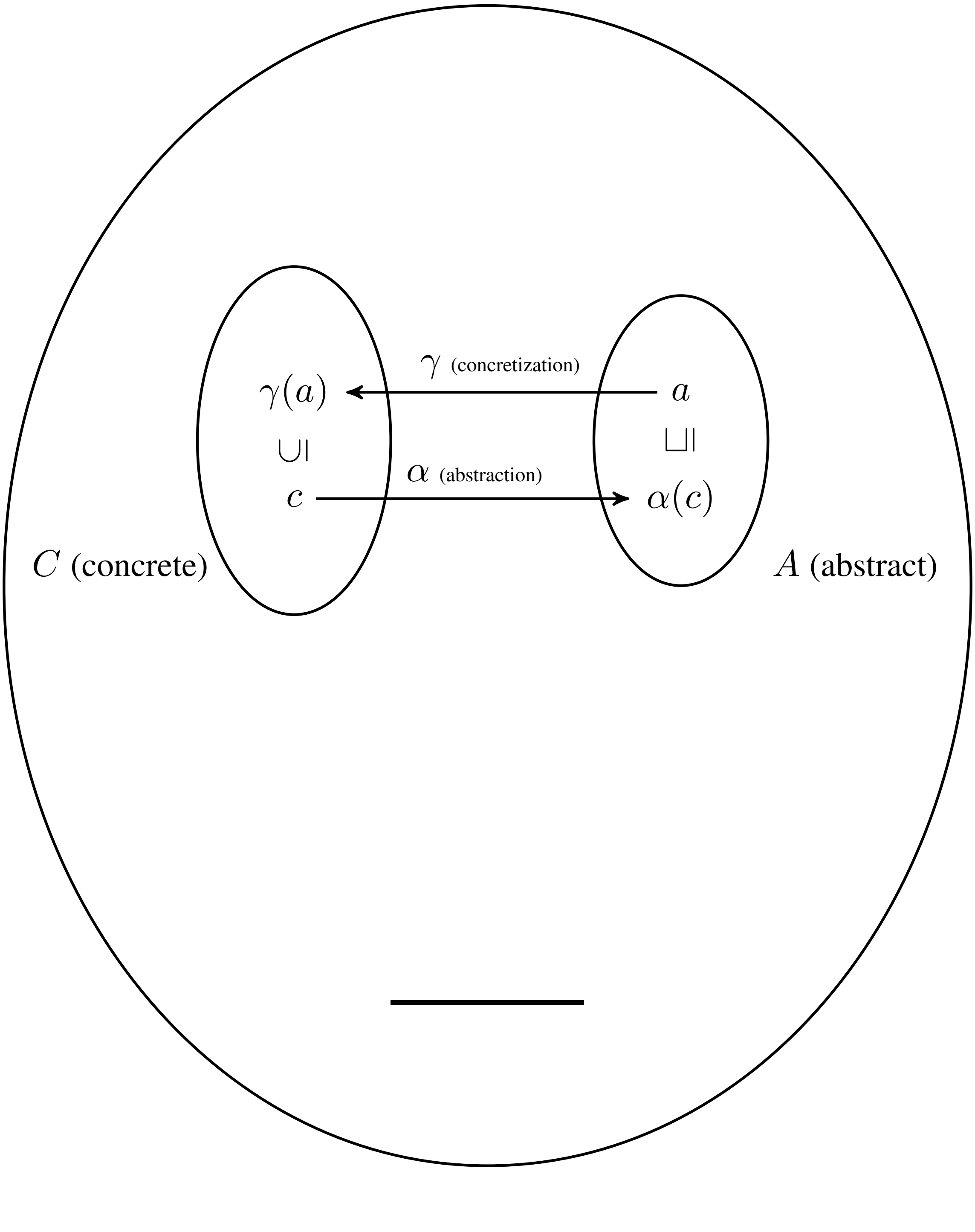
Quiz Solutions Outline

Synthesis, Analysis, and Verification 2013

for the quiz given on Friday, May 3rd, 2013



Problem 1: Recursion ([20 points])

Task a) [5 points]
Because the state has three components, relations on states contain tuples $((r, x, y), (r', x', y')) \in \mathbb{Z}^6$. Sets of such states are elements of $2^{\mathbb{Z}^6}$. The type of functions mapping such relations to new relations is $E : 2^{\mathbb{Z}^6} \to 2^{\mathbb{Z}^6}$.
If ρ maps a command into a relation, then the definition of function E is:

$$\begin{aligned} E(r_f) =& \Big(\Delta_{S(y<0)} \circ \rho(y' = y+1) \circ r_f \circ \rho(r' = r - x)\Big) \cup \\ & \Delta_{S(y\geq 0)} \circ \Big(\Delta_{S(y\neq 0)} \circ \rho(y' = y-1) \circ r_f \circ \rho(r' = r + x)\Big) \cup \Delta_{S(y=0)} \end{aligned}$$

We can also substitute the meaning of assignments $\rho(v = e) \equiv \quad v' = e \bigwedge_{w\neq v} w' = w$, which gives $E(r_f)$ to be

$$\begin{aligned} &\Big(\Delta_{S(y<0)} \circ \{...|y' = y+1 \wedge x' = x \wedge r' = r\} \circ r_f \circ \{...|r' = r - x \wedge x' = x \wedge y' = y\}\Big)\Big) \cup \\ &\Delta_{S(y\geq 0)} \circ \Big(\Delta_{S(y\neq 0)} \circ \{...|y' = y-1 \wedge x' = x \wedge r' = r\} \circ r_f \circ \{...|r' = r + x \wedge x' = x \wedge y' = y\} \cup \Delta_{S(y=0)}\Big) \end{aligned}$$

where "..." in the above expression denotes the part of the comprehebsion $((r, x, y), (r', x', y'))$.

Task b) [10 points] We simplify further the derived definition of $E(r_f)$ using the definition of relation composition and union:

$$\begin{aligned} E(\sigma) =& \{((r, x, y), (r', x', y'))| \quad \exists r_1, x_1, y_1, r_2, x_2, y_2.\ y < 0 \wedge r_1 = r \wedge x_1 = x \wedge y_1 = y+1 \wedge \\ & \qquad r_2 = r_1 + x_1 * y_1 \wedge r' = r_2 - x_2 \wedge y' = y_2 \wedge x' = x_2\} \cup \\ & \{((r, x, y), (r', x', y'))| \quad \exists r_1, x_1, y_1, r_2, x_2, y_2 . y > 0 \wedge r_1 = r \wedge x_1 = x \wedge y_1 = y-1 \wedge \\ & \qquad r_2 = r_1 + x_1 * y_1 \wedge r' = r_2 + x_2 \wedge x' = x_2 \wedge y' = y_2\} \cup \\ & \{((r, x, y), (r', x', y'))| \quad y = 0 \wedge r' = r \wedge y' = y \wedge x' = x)\} \end{aligned}$$

We next eliminate quantifiers:

$$\begin{aligned} E(\sigma) =& \{((r, x, y), (r', x', y'))| \quad \exists x_2.\ y < 0 \wedge r' = r + x * (y+1) - x_2\} \cup \\ & \{((r, x, y), (r', x', y'))| \quad \exists x_2 . y > 0 \wedge r' = r + x * (y-1) + x_2\} \cup \\ & \{((r, x, y), (r', x', y'))| \quad y = 0 \wedge r' = r \wedge y' = y \wedge x' = x\} \end{aligned}$$

We see that we can pick x_2 arbitrarily, hence if we pick $x_2 \neq x$, $E(\sigma) \subsetneq \sigma$.

Another much simpler solutions is:

$$z = ((0, 1, 0), (42, 1, 42)) \in E(\sigma) \quad \text{but not } z \in \sigma$$

Task c) [5 points]
Let $s = \{((r, x, y)(r', x', y')) \mid r' = r + x * y \wedge x' = x\}$

$$\begin{aligned} E(s) =& \{((r, x, y), (r', x', y')) \mid \quad \exists r_1, x_1, y_1, r_2, x_2, y_2.\ y < 0 \wedge r_1 = r \wedge x_1 = x \wedge y_1 = y+1 \wedge \\ & \qquad r_2 = r_1 + x_1 * y_1 \wedge x_2 = x_1 \wedge r' = r_2 - x_2 \wedge y' = y_2 \wedge x' = x_2\} \cup \\ & \{((r, x, y), (r', x', y')) \mid \quad \exists r_1, x_1, y_1, r_2, x_2, y_2 . y > 0 \wedge r_1 = r \wedge x_1 = x \wedge y_1 = y-1 \wedge \\ & \qquad r_2 = r_1 + x_1 * y_1 \wedge x_2 = x_1 \wedge r' = r_2 + x_2 \wedge x' = x_2 \wedge y' = y_2\} \cup \\ & \{((r, x, y), (r', x', y')) \mid \quad (r' = r \wedge y' = y \wedge x' = x)\} \end{aligned}$$

Eliminating quantifiers:

$$\begin{aligned}E(s) =&\{((r,x,y),(r',x',y')) \mid y<0 \wedge r' = r + x*(y+1) - x = r + x*y \wedge x' = x\}\cup\\ &\{((r,x,y),(r',x',y')) \mid y>0 \wedge r' = r + x*(y-1) + x = r + x*y \wedge x' = x)\}\cup\\ &\{((r,x,y),(r',y')) \mid y=0 \wedge r' = r + x*y \wedge y' = y \wedge x' = x\}\end{aligned}$$

We have thus shown that $E(s) \subseteq s$. Then since we know that the least fixpoint z satisfies $E(z) = z$, we know that $z \subseteq s$, and hence the specification is satisfied.

Problem 2: Transitive closure ([20 points])

Task a) [5 points]

$$\begin{aligned}
sp(sp(P,r^*),r) &= \{s'|\exists s.s \in sp(P,r^*) \wedge (s,s') \in r\} \\
&= \{s'|\exists s.s \in \{t'|\exists t.t \in P \wedge (t,t') \in r^*\} \wedge (s,s') \in r\} \\
&= \{s'|\exists s,t.t \in P \wedge (t,s) \in r^* \wedge (s,s') \in r\} \\
&= \{s'|\exists t.t \in P \wedge (t,s') \in r^* \circ r\} \\
&= sp(P,r^+) \subseteq sp(P,r^*) \subseteq S
\end{aligned}$$

Task b) [10 points] Let us call I_0 the condition that holds after executing r_1.

$$\begin{aligned}
I_0 = sp(P, x' = 4*x \wedge y' = x+3) &= \{(x',y')|\exists x,y.x \geq 0 \wedge y \leq -5 \wedge x' = 4*x \wedge y' = x+3\} \\
&= \{(x',y')|y' \geq 3 \wedge x' \geq 0 \wedge 4y' = x'+12 \wedge 4|x'\}
\end{aligned}$$

The formula corresponding to $r_2 \circ r_3$ is given by

$$\begin{aligned}
&\exists x_1, y_1.x_1 = y \wedge y_1 = x+1 \wedge x' = y_1+1 \wedge y' = x_1 \\
\Leftrightarrow & y' = y \wedge x' = x+2
\end{aligned}$$

From the lectures we know that the transitive closure for $x' = x+2$ is

$$\exists k.k \geq 0 \wedge x' = x + 2*k \wedge y' = y$$

Then, I_1 is given by $I_1 = I_0 \cup sp(I_0, r^*)$.

$$\begin{aligned}
sp(I_0,r^*) &= \{((x,y),(x',y'))|\exists x,y.y \geq 3 \wedge x \geq 0 \wedge 4|x \wedge 4y = x+12 \wedge \exists k.k \geq 0 \wedge x' = x+2*k \wedge y' = y\} \\
&= \{((x,y),(x',y'))|\exists x.y' \geq 3 \wedge x \geq 0 \wedge 4|x \wedge 4y' = x+12 \wedge x'-x \geq 0 \wedge 2|x'-x\}
\end{aligned}$$

Thus, $I_1 = y \geq 3 \wedge 2|x-(4y-12) \wedge x \geq 4y-12$.
We compute I_2 using the strongest precondition again:

$$\begin{aligned}
sp(I_2,r_4) &= \{(x',y')|\exists x,y.y \geq 3 \wedge 2|x-(4y-12) \wedge x \geq 4y-12 \wedge x' = y-x \wedge y' = y\} \\
&= \{(x',y')|y' \geq 3 \wedge 2|-3y'-x'+12 \wedge -x' \geq 3y-12\}
\end{aligned}$$

Thus, $I_2 = y \geq 3 \wedge 2|-3y-x+12 \wedge -x \geq 3y-12$

Task c) [5 points]

- r_2 and r_3 are difference bounds relations, for which we know from the lectures the transitive closure is expressible in Presburger arithmetic.
- Then, Presburger arithmetic admits quantifier elimination which allows us to obtain quantifier-free expressions.

Problem 3: Hoare logic ([20 points])

Task a) [4 points]

$$\forall i, j, v, w.(i, v) \in L \land (j, w) \in L \rightarrow v = w$$
$$\forall i, v.(i, v) \in L \rightarrow i \geq 0$$

Task b) [4 points] $k = 0 \land S = S_0 \land L = \emptyset \land \forall v.v \in S \rightarrow v \geq 0$
Task c) [12 points] Invariant:

$$A : \forall v.v \in S_0 \rightarrow (v \in S \lor \exists i.(i, v) \in L)$$
$$B : \forall i, j.(i, v) \in L \land (j, w) \in L \land i < j \rightarrow v < w < k$$

i) Before the loop, we have $L = \emptyset$ so that condition B holds trivially and condition A reduces to $\forall v.v \in S_0 \rightarrow v \in S_0$ which also trivially holds.

ii) Now we need to show that invariant is inductive. That is we need to show the following implication holds:

$$\begin{aligned} &\forall v.v \in S_0 \rightarrow (v \in S \lor \exists i.(i, v) \in L)\land \qquad (*) \\ &\quad \forall i, j.(i, v) \in L \land (j, w) \in L \land i < j \rightarrow v < w < k \land \text{loop body} \\ &\quad \rightarrow \\ &\forall v.v \in S_0 \rightarrow (v \in S' \lor \exists i.(i, v) \in L')\land \\ &\quad \forall i, j.(i, v) \in L' \land (j, w) \in L' \land i < j \rightarrow v < w < k' \end{aligned}$$

We consider two cases. In the first case, when $k \notin S$, then the loop body is

$$S \neq \emptyset \land k' = k + 1$$

and we see that the implication * holds, since if $v < w < k$ then also $v < w < k + 1$.

In the second case the loop body is the following:

$$\begin{aligned} S \neq \emptyset \land k \in S \ \land L' = L \cup \{(size(L), k)\} \land \\ S' = S \setminus \{k\} \ \land k' = k + 1 \end{aligned}$$

Substituting for the primed values into *:

$$\forall v.v \in S_0 \rightarrow (v \in S \lor \exists i.(i, v) \in L)\land \tag{1}$$
$$\forall i, j.(i, v) \in L \land (j, w) \in L \land i < j \rightarrow v < w < k \land \text{loop body} \tag{2}$$
$$\rightarrow \tag{3}$$
$$\forall v.v \in S_0 \rightarrow (v \in (S \setminus \{k\}) \lor \exists i.(i, v) \in (L \cup \{(size(L), k)\}))\land \tag{4}$$
$$\forall i, j.(i, v) \in (L \cup \{(size(L), k)\}) \land (j, w) \in (L \cup \{(size(L), k)\}) \land i < j \rightarrow v < w < k + 1 \tag{5}$$

From line 1, line 4 holds for all elements in S_0 except for k, which is now removed from S. But since there exists $i = size(L)$ such that $(size(L), k) \in L$, the condition on line 4 holds.

From line 2, line 5 holds for all i, j, except when $j = size(L)$. But when $i, j < size(L)$, then we know from the assumption that $v < w < k$. Then if $j = size(L)$, $w = k$ and thus w strictly larger than any v, thus the condition still holds.

Hence, we have shown that invariant holds after one loop iteration is thus inductive.

iii) After the loop we have $S = \emptyset$ so that condition A becomes $\forall v.v \in S_0 \rightarrow \exists i.(i, v) \in L$ and condition B implies immediately the first part of the postcondition.

Problem 4: Galois connection ([20 points])

Task a) [5 points] We will prove that (α, γ) is a Galois connection. To show this, we will show $c \subseteq \gamma(a) \Leftrightarrow \alpha(c) \sqsubseteq a$. Since the ordering on the abstract domain is the superset relation, this becomes

$$c \subseteq \gamma(a) \Leftrightarrow \alpha(c) \supseteq a \quad \text{i.e.} \quad c \subseteq \gamma(a) \Leftrightarrow a \subseteq \alpha(c)$$

$$\begin{aligned}
c \subseteq \gamma(a) &\Leftrightarrow \forall s. s \in c \to s \in \gamma(a) \\
&\Leftrightarrow \forall s. s \in c \to \forall t \in a.\ (s,t) \in r \\
&\Leftrightarrow \forall s \in c. \forall t \in a.\ (s,t) \in r \\
&\Leftrightarrow \forall t \in a. \forall s \in c.\ (s,t) \in r \\
&\Leftrightarrow \forall t. t \in a \to \forall s \in c.\ (s,t) \in r \\
&\Leftrightarrow a \subseteq \alpha(c)
\end{aligned}$$

Task b) [4 points] No. Let $S = T = \{a, b, c, d\}$ and $r = \{(a,a), (b,a), (c,c), (c,d)\}$.
Then $\gamma(\{a\}) = \gamma(\{a,b\}) = \{a\}$, so γ is not injective. Neither is it surjective as the element b is never mapped to any subset of S.
Conversely, $\alpha(\{c\}) = \alpha(\{c,d\}) = \{c\}$, so α is not injective. Neither is it surjective as the element d is never mapped to any subset of T.

Task c) [5 points] Yes, this is a Galois connection and it corresponds to predicate abstraction.

Task d) [6 points]

$$\begin{aligned}
\gamma(Q) &= \{s \in S \mid \forall t \in Q.(s,t) \in \overline{\rho(z)}\} \\
&= \{s \in S \mid \forall t. t \in Q \to (s,t) \in \overline{\rho(z)}\} \\
&= \{s \in S \mid \forall t.(s,t) \in \rho(z) \to t \in \overline{Q}\} \\
&= wp(\rho(z), \overline{Q})
\end{aligned}$$

$$\begin{aligned}
\alpha(P) &= \{t \in T \mid \forall s \in P.(s,t) \in \overline{\rho(z)}\} \\
&= \{t \in T \mid \forall s. s \in P \to (s,t) \in \overline{\rho(z)}\} \\
&= \{t \in T \mid \neg \exists t. s \in P \land (s,t) \in \rho(z)\} \\
&= \overline{sp(P, \rho(z))}
\end{aligned}$$

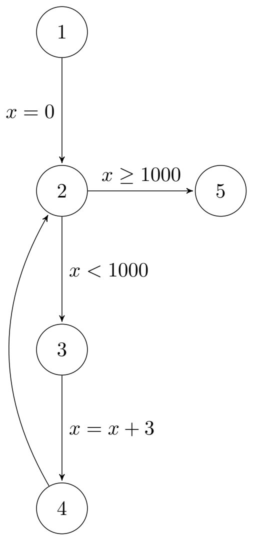
$$\begin{gathered}
c \subseteq \gamma(a) \Leftrightarrow \alpha(c) \supseteq a \\
c \subseteq wp(\rho(z), \overline{Q}) \Leftrightarrow \overline{sp(P, \rho(z))} \supseteq a \\
c \subseteq wp(\rho(z), \overline{Q}) \Leftrightarrow sp(P, \rho(z)) \subseteq \overline{a}
\end{gathered}$$

Problem 5: Widening ([20 points])

Task a)
1) [2 points] By definition of Galois connection, α and γ are monotonic. A composition of two monotonic functions is monotonic. Indeed, say $c_1 \subseteq c_2$. Then $\alpha(c_1) \subseteq \alpha(c_2)$ by monotonicity of α. Furthermore, then $\gamma(\alpha(c_1)) \subseteq \gamma(\alpha(c_2))$ by monotonicity of γ. Therefore, α' is monotonic.
2) [2 points]

Task b) [3 points] The type signature of α' is $\alpha' : C \to C$ i.e. $2^{\mathbb{Z}} \to 2^{\mathbb{Z}}$.
As an example, let $c = \{5, 10, 15\}$. Then $\alpha' = \{x \mid 0 \leq x \leq 100\}$.
Task c) [5 points] Note that the image of α' is isomorphic to the lattice $(A, \sqsubseteq)$ and the image of $\overline{\alpha}'$ is isomorphic to A^n. Iterating $\overline{\alpha}'$ is like iterating an abstract transformer in A^n. The longest chain in A has length 7. With n program points the number of steps us $7n$, so we can take $H = 7n$.

Task d) [6 points] The program can have the following control-flow graph:



The fixpoint of $\overline{\alpha}'$ at the control-locations is then:

$$
\begin{aligned}
1 :& \mathbb{Z} \\
2 :& \{0, \ldots, \infty\} \\
3 :& \{0, \ldots, 1000\} \\
4 :& \{0, \ldots, \infty\} \\
5 :& \{1000, \ldots, \infty\}
\end{aligned}
$$

After applying F again we get:

$$
\begin{aligned}
1 :& \mathbb{Z} \\
2 :& \{0, \ldots, \infty\} \\
3 :& \{0, \ldots, 999\} \\
4 :& \{3, \ldots, 1003\} \\
5 :& \{1000, \ldots, \infty\}
\end{aligned}
$$