Homework 2

March 4, 2011

Problems 1 through 3 are due on March 11. Only Problem 4 is due on March 14.

1 Problem 1 (pen and paper)

Let r be an arbitrary relation. Prove that $r^{-1} \circ (r \cup r^{-1})^* \circ r$ is symmetric.
Hint: consider what it means to have $(x, y) \in r^*$ for some concrete (x, y).

2 Problem 2 (pen and paper)

Let $E(r_1, r_2, ..., r_n)$ be a relation composed of relations r_i with the operators

- $\circ$ (sequential composition)
- $\cup$ (nondeterministic choice)
- $r1 \setminus r2$ (set difference)
- $r1 \prec r2 = \{(x, y) \mid ref(r1, r2, x) \rightarrow y = x\}$ where $ref(r1, r2, x)$ denotes the condition $\forall z.\ ((x, z) \in r_1 \rightarrow (x, z) \in r_2)$

Assume that each relation r_i occurs exactly once in such an expression $E(r_1, r_2, ..., r_n)$. Describe an algorithm to compute, for each relation in the expression (represented as a tree), whether it is monotonic or anti-monotonic with respect to that relation.

Notes:

- The intuition for the last operator is that ref is a kind of "local" relation inclusion, and $r1 \prec r2$ is an assertion that fails if, in the current state, executing r1 would give more behaviors than executing r2.
- The expression being anti-monotic means that

$$r_i \subseteq r_i' \rightarrow E(r_1, r_2, ..., r_i, ..., r_n) \supseteq E(r_1, r_2, ..., r_i', ..., r_n)$$

3 Problem 3 (on paper)

Recall the guarded command language from Lecture 2 and consider the constructs for sequential composition and the if-statement:

$$\begin{array}{rcl} \text{s1; s2} & \rightsquigarrow & r_{s1} \circ r_{s2} \\ \text{(assume(F); s1) [] (assume(F); s2)} & \rightsquigarrow & (\triangle_F \circ s_1) \cup (\triangle_{\neg F} \circ s_2) \end{array}$$

Show that if `s1` and `s2` are deterministic statements (i.e. functions), then their composition using

(i) sequential composition

(ii) if-statement

remains deterministic.

From this, prove by induction on syntax tree that relational expressions built from functions using sequential composition and 'if' statement remains a function.

You can use the same style of proof as in Problem 5 from Exercises 2.

4 Problem 4 (in Isabelle)

Formalize the proof of Problem 3 in Isabelle. Use the provided Isabelle file and complete the missing parts. To help you, we also provide a complete proof of a simpler fact. During proof development, feel free to use sledgehammer. Your final proof can contain e.g. metis and auto, but should not contain the 'sorry' command.