SAV'07 Homework 1: Due before the start of the lecture on Thursday, 22 March 2007

March 18, 2007

Note: Feel free to do problems in groups of up to 3 people. Please write on top of your solutions who you were collaborating with on the homework.

1 More properties of relations

For each of the following properties, either prove it or show a counterexample relations for which it does not hold:

1. $r \circ (s \cap t) \subseteq (r \circ s) \cap (r \circ t)$
2. $(r \circ s) \cap (r \circ t) \subseteq r \circ (s \cap t)$

Define r^{-1}, the inverse of a relation r by

$$r^{-1} = \{(y, x) \mid (x, y) \in r\}$$

Prove 3. $(r \circ s)^{-1} = s^{-1} \circ r^{-1}$.

2 Weakest preconditions

Weakest preconditions are a way of propagating correctness conditions starting from the end of a program towards the beginning.

First, define the notion of a Hoare triple, denoted $\{P\}r\{Q\}$ as follows:

$$\{P\}r\{Q\} \leftrightarrow (\forall s_1, s_2.\ (s_1 \in P \wedge (s_1, s_2) \in r)\ \rightarrow\ s_2 \in Q)$$

The condition $\{P\}r\{Q\}$ means that if the initial state belongs to P, then the final state after executing r belongs to Q.

The *weakest precondition* of Q with respect to r, denoted $\mathrm{wp}(r, Q)$ is the largest set P such that $\{P\}r\{Q\}$.

Prove 4. $\mathrm{wp}(r, Q) = \{s_1 \mid \forall s_2.\ ((s_1, s_2) \in r \rightarrow s_2 \in Q)\}$.

Do the following conditions hold for each relation r and sets Q_1, Q_2:

5. $\mathrm{wp}(r, Q_1 \cup Q_2) = \mathrm{wp}(r, Q_1) \cup \mathrm{wp}(r, Q_2)$
6. $\mathrm{wp}(r, Q_1 \cap Q_2) = \mathrm{wp}(r, Q_1) \cap \mathrm{wp}(r, Q_2)$

3 Formula evaluation and substitution

Note: Please try to hand in this problem at the same time as the other ones. However, because these are the first implementation problems, if you need more time, please let me know and you can get an extension for the problems below.

Write a data structure (abstract syntax tree) for representing formulas whose syntax conforms to the following grammar.

$$
\begin{aligned}
&v ::= x \mid y \mid z \\
&t ::= v \mid 0 \mid 1 \mid 2 \mid t + t \mid t - t \\
&F ::= t = t \mid t < t \mid F \wedge F \mid F \vee F \mid \neg F \mid \exists x < t.F \mid \forall x < t.F
\end{aligned}
$$

Here x, y, z range over *non-negative integers*. Note that we used bounded quantifiers: $\exists x < t.F$ means that there exists a non-negative integer smaller than t such that F holds, whereas $\forall x < t.F$ means that F holds for all values of x that are smaller than the value of t.

7. Implement a procedure that accepts as input an abstract syntax tree F of a formula, as well as an interpretation $e : \{x, y, z\} \to \{0, 1, 2, \ldots\}$, which maps variable names x, y, z to non-negative integers. Your procedure should output the truth value $[\![F]\!]e$, which is either "true" or "false".

Your implementation should be based on the semantics of first-order logic, for example,

$$[\![F_1 \wedge F_2]\!]e = [\![F_1]\!]e \wedge [\![F_2]\!]e$$

You will also need to implement an evaluation function on terms, which computes the value $[\![t]\!]e$ of a term in a given interpretation.

You can implement parsing and pretty printing of formulas if it helps you debug your function. In any case, provide at least three test cases for your function (for example, you can write code that constructs some example abstract syntax trees of formulas as well as the mappings e and then invokes the evaluation function).

8. Implement a procedure that takes a formula F, a variable name such as x, and an arithmetic term t, and outputs a new formula F_1, which is the result of replacing all occurrences of x in F by t. Test your procedure on at least three examples one of which is the formula

$$\forall x < 4.(x < 2 \vee (\exists y < x + x.x < y))$$

9.(optional) Denote F_1 from previous task by $F[x := t]$. Does the following property hold:

$$[\![F[x := t]]\!]e \;\leftrightarrow\; [\![F]\!](e[x \mapsto [\![t]\!]e])$$

where

$$e[x \mapsto v](z) = \begin{cases} v, & z = x \\ e(z), & z \neq x \end{cases}$$

Prove or explain informally why you think this is the case.