CS734 Assignment 1: Typechecking, Tables & Functional Software Verification

Due: 1330 Monday February 14, 2002

All of your PVS work for this assignment should be done in a single file called `a1.pvs`. Create a new theory for each question with the following naming scheme. The theory containing all work for question 1 should be called `A1Q1`, the theory for question 2 should be called `A1Q2`, etc. Please check the course website for information on how to submit the PVS part of your work electronically. You can hand in your written work at the start of class on the due date.

1. **Basic Propositional Logic in PVS** (15 marks)

   Consider the following simple theorem of propositional logic:

   $$(p \wedge q \rightarrow r \vee s) \leftrightarrow (p \rightarrow (q \rightarrow r \vee s))$$

   a) Using only the proof rules from slides 3 and 4 of the *Sequent Calculus and PVS* section of the notes, do a formal proof of the above formula.

   b) In PVS prove that the above using the `(FLATTEN)` and `(SPLIT)` commands.

   c) In PVS prove that the above using the `(BDDSIMP)` command.

2. **PVS Basics: Predicate Logic in PVS - Part 1** (15 Marks Total)

   a) (5 marks) Using PVS, you will determine if the following set of premises is consistent or inconsistent.

   $$\Gamma := \{\forall x(P(x) \rightarrow \exists y Q(x,y)), \neg[\exists x \neg P(x) \vee \exists y Q(y,y)]\}$$

   b) (5 marks) In the file `a1.pvs`, create a theory called `A1Q2b` write down a formula called `Q2b` that you would try to prove to show that the premises are inconsistent. Try to prove this formula.

   c) (5 marks) Create a new theory called `A1Q2c`. In this theory, create an interpretation structure (i.e., a model) that satisfies all of the premises and then state a theorem called `Q2c` that you can prove to show that all of the premises are true. Prove this theorem.

3. **PVS Basics: Predicate Logic in PVS - Part 2** (10 Marks Total)

   a) (5 marks) Determine if the following argument is valid or invalid by attempting to prove a theorem called `Q3a` in a PVS theory called `A1Q3`.

   Premises: $\forall x(R(a,x) \rightarrow a = x \vee a = b), \exists x R(a,x), S(a) \wedge \neg S(b)$

   Conclusion: $R(a,a)$ Here x is a variable while a and b are constants.

   b) (5 marks) Determine if the following set of premises is consistent or inconsistent by proving an appropriate theorem in PVS called `Q3b`.

   $$\Gamma := \{\exists x \forall y(x = y), \exists x \exists y(x \neq y)\}$$

4. **Partial Functions, Types & Predicate Subtypes in Logic** (10 Marks Total)

   a) (3 marks) Consider the function:

   $$f(x) = \ln x$$

   In your PVS file, write down the best PVS definition for f. Make sure there are no unproved TCCs resulting from the definition. (Define the ln (natural logarithm) function as an uninterpreted function.)

b) (7 marks) Using the definition from part (a), create the "best" PVS definitions for the function: $f(x, y) = \ln(x - (y + 1)^2)$ Again make sure there are no unproved TCCs resulting from the definition.

5. **Tabular Specification I: Weakening conditions on tabular definitions** (30 Marks Total)

Consider the tabular definition:

$$f(x, y) = \begin{array}{|c|c|c|} \hline C_1(x, y) & C_2(x, y) & C_3(x, y) \\ \hline f_1(x, y) & f_2(x, y) & f_3(x, y) \\ \hline \end{array}$$

a) (6 marks) Let U be the nonempty universe that variables x, y range over. For the table to be properly specified, it need not be the case that each of the functions is of type $U \times U \to U$. We need only assume functions f_1, f_2 and f_3 are defined when C_1, C_2 and C_3 are respectively true. Create a PVS definition for this table where the conditions C_i are uninterpreted two place predicates over U (i.e. they have type $U \times U \to bool$) and each of the 2-ary functions is an uninterpreted function with its domain restricted to the predicate subtype defined by the predicate at the top of its column (i.e. the domain of f_1 is given by the predicate subtype defined by C_1).

Be sure that your definition type checks. You should be able to prove all TCCs except for the two associated with the Disjointness and Completeness of the table.

b) (6 marks) Add the definitions contained in Figure 1 to your `Q1` Theory. Typecheck the theory and

```
x: VAR real
                    %--------------%
 g(x): real = TABLE |[ x<0  | x>=0 ]|
                    %--------------%
                    |  x    | 2*x  ||
           ENDTABLE %--------------%

                    %---------------%
 h(x): real = TABLE |[ x<=0 | x>=0 ]|
                    %---------------%
                    |   x   | 2*x  ||
           ENDTABLE %---------------%

same: THEOREM g=h
```

Figure 1: Disjointness condition counter example

have PVS try to prove the resulting TCCs with the PVS/Parsing and Typechecking/typecheck-prove command. Take a look at the TCCs for the file with the PVS/Viewing TCCs/show-tccs command. As you can see, the disjointness condition `h_TCC1` fails for the table defining `h`. Rerun the proof PVS tried by placing you cursor on this TCC and invoking the prover. This is a case where the table still defines a total function even though PVS' disjointness condition is violated. Why?

c) (6 marks) The PVS example in Figure 1 provides some insight as to why the Disjointness condition generated by PVS is overly restrictive. The theorem "same" can be easily proved using the (GRIND) command. For the table used to define f in part (a) above, what is a weaker "disjointness" condition that together with the completeness condition provides necessary and sufficient conditions for the table defining f to be a total function.

d) (2 marks) Although the weakened "disjointness" condition together with the usual completeness condition provides necessary and sufficient conditions for a table to define a function. Why is it preferable for software engineers to use the more strict disjointness condition when using tables to specify the functional requirements of software?

e) (10 marks) Use PVS to prove that your weaker disjointness condition together with the completeness condition provide necessary and sufficient conditions for f to be a function.