universität innsbruck

Program Verification	SS 2023	LVA 703083+703084

Sheet 9	Deadline: May 24, 2023, 10am

- Prepare your solutions on paper.
- Mark the exercises in OLAT before the deadline.
- Marking an exercise means that a significant part of that exercise has been treated.

Exercise 1 *Polynomial Interpretations* **5 p.**

Consider the following functional program for computing the binary logarithm.

$$\begin{aligned}
\mathsf{half}(\mathsf{Zero}) &= \mathsf{Zero}\\
\mathsf{half}(\mathsf{Succ}(\mathsf{Zero})) &= \mathsf{Zero}\\
\mathsf{half}(\mathsf{Succ}(\mathsf{Succ}(x))) &= \mathsf{Succ}(\mathsf{half}(x))\\
\mathsf{log2}(\mathsf{Zero}) &= \mathsf{Zero}\\
\mathsf{log2}(\mathsf{Succ}(\mathsf{Zero})) &= \mathsf{Zero}\\
\mathsf{log2}(\mathsf{Succ}(\mathsf{Succ}(x))) &= \mathsf{Succ}(\mathsf{log2}(\mathsf{Succ}(\mathsf{half}(x))))
\end{aligned}$$

1. Write down all dependency pairs that cannot be solved by the subterm criterion and determine the usable equations for these dependency pairs. (1 point)
2. Prove termination via polynomial interpretations. First setup the constraints symbolically, and then choose between manual solving and SMT-solving. For the latter you can either directly download and compile Z3 from github, or use a binary version that is distributed as part of Isabelle in the `contrib/z3...` directory. (4 points)

Exercise 2 *Usable Equations* **15 p.**

The usable equations can be refined to usable equations w.r.t. an argument filter where an argument filter describes for each function symbol which arguments of that symbol are considered and which are ignored by the reduction pair.
Formally, an argument filter is a map π such that $\pi(F) \subseteq \{1, \dots, n\}$ for every n-ary function symbol F.
The idea is now to refine usable equations in a way that only those defined symbols are considered that appear on positions in right-hand sides which are not ignored.
For instance, in an dependency pair $\mathsf{f}^\sharp(\mathsf{Cons}(x, xs), \mathsf{Succ}(y)) \to \mathsf{f}^\sharp(\mathsf{reverse}(xs), \mathsf{g}(x, y))$ we might use $\pi(\mathsf{f}^\sharp) = \{1\}$ and then only get usable equations for reversal, but not for function g. This should lead to a successful termination proof since the first argument gets shorter, and it does not matter what is happening in the second argument, i.e., it does not matter how complex the g-equations are.

1. Provide a definition of usable equations w.r.t. a fixed argument filter π. In particular, there should be a definition of $\mathcal{U}(t)$, the usable equations of a term, where t is a (subterm) of a right-hand side. You can assume that $\mathcal{E}_f$ is the set of equations of the program that defines f. (3 points)
2. Provide a definition that a reduction pair $(\succ, \succsim)$ is compatible with an argument filter, i.e., that whenever $i \notin \pi(F)$, then the i-th argument of F must be ignored by $\succsim$. (1 point)
3. Prove soundness of usable rules: whenever $\ell \succsim r$ for all usable equations of $\ell = r \in \mathcal{U}(t)$ and $t\sigma \overset{i}{\hookrightarrow}{}^* s$ for some normal-form substitution σ, then $t\sigma \succsim s$. W.l.o.g. you may assume that $\succsim$ is reflexive. (6 points)

4. One problem in the automation of usable equations w.r.t. and argument filter is that we do not want to fix π in advance, i.e., we want to avoid having to iterate over all possible choices of π. Therefore, one usually also treats π symbolically, i.e., "$i \in \pi(F)$" will be seen as a Boolean variable in the SMT-encoding for polynomials. Similarly, one adds Boolean variables u_f to indicate that the f-equations are usable.

   Using these ideas, design an encoding to search for linear polynomial interpretations and apply it on the example set of dependency pairs P from above and the following equations. You can just write $encode(n)$ for the SMT-encoding of $\ell \succsim r$ where n corresponds to the number of the equation $\ell = r$. (5 points)

$$
\begin{aligned}
\mathsf{reverse}(xs) &= \mathsf{rev}(xs, \mathsf{Nil}) && (1)\\
\mathsf{rev}(\mathsf{Nil}, ys) &= ys && (2)\\
\mathsf{rev}(\mathsf{Cons}(x, xs), ys) &= \mathsf{rev}(xs, \mathsf{Cons}(x, ys)) && (3)\\
\mathsf{g}(x, y) &= \mathsf{double}(\mathsf{incr}(x)) && (4)\\
\mathsf{double}(\mathsf{Zero}) &= \mathsf{Zero} && (5)\\
\mathsf{double}(\mathsf{Succ}(x)) &= \mathsf{Succ}(\mathsf{Succ}(\mathsf{double}(x))) && (6)\\
\mathsf{incr}(x) &= \mathsf{Succ}(x) && (7)
\end{aligned}
$$