Exercise 13.1  Sign Analysis

Instantiate the abstract interpretation framework to a sign analysis over the lattice \( \textit{pos}, \textit{zero}, \textit{neg}, \textit{any} \), where \( \textit{pos} \) abstracts positive values, \( \textit{zero} \) abstracts zero, \( \textit{neg} \) abstracts negative values, and any abstracts any value.

For this exercise, you best modify the parity analysis \texttt{src/HOL/IMP/Abs_Int1_parity}.

Exercise 13.2  Complete Lattices: GLB of UBs is LUB

Show that the least upper bound is the greatest lower bound of all upper bounds:

\[
\text{definition} \quad "\text{Sup}' (S::'a::complete_lattice set) \equiv \text{Inf}\{u. \forall s\in S. s\leq u\}"
\]

\[
\text{lemma} \quad \text{Sup}'\_\text{upper}: \quad \forall s\in S. s \leq \text{Sup}' S
\]

\[
\text{lemma} \quad \text{Sup}'\_\text{least}:
\begin{align*}
\text{assumes} & \quad \text{upper}: \quad \forall s\in S. s\leq u \\
\text{shows} & \quad \text{"Sup}' S \leq u"
\end{align*}
\]

Homework 13.1  Lattice Theory

\textit{Submission until Tuesday, 07.02.2017 (Pen & Paper), 10:00am.}

A type \( 'a \) is a \( \sqcup \)-semilattice if it is a partial order and there is a supremum operation \( \sqcup \) of type \( 'a \Rightarrow 'a \Rightarrow 'a \) that returns the least upper bound of its arguments:

\begin{itemize}
  \item Upper bound: \( x \leq x \sqcup y \) and \( y \leq x \sqcup y \)
  \item Least: \( x \leq z \land y \leq z \rightarrow x \sqcup y \leq z \)
\end{itemize}

Is every finite \( \sqcup \)-semilattice with a bottom element \( \bot \) also a complete lattice? Prove or disprove on paper!
Homework 13.2  AI for the Extended Reals

Submission until Tuesday, 07.02.2017 (Isabelle), 10:00am. For this exercise, we will consider a modified variant of IMP that computes on real numbers extended with $-\infty$ and $\infty$. The corresponding type is ereal. We will consider “$-\infty + \infty$” and “$\infty + (-\infty)$” erroneous computations. We propagate errors by using the option type, i.e. we set $val = \text{ereal option}$. The files AExp.thy, BExp.thy, BigStep.thy, Collecting.thy for this variant are provided for you on the website. Your task is now to design an abstract interpreter on the domain consisting of subsets of $\{\infty^-, \infty^+, \text{NaN}, \text{Real}\}$ where NaN signals a computation error and all other values have their obvious meaning. First adopt Abs_Int0.thy and Abs_Int1.thy to accommodate for the changed semantics, and then instantiate the abstract interpreter (Abs_Int, Abs_Int_mono, and Abs_Int_measure) with your analysis. For this step you best modify the parity analysis Abs_Int1_parity.thy.

Hints: To benefit from proof automation it can be helpful to slightly change the format of the rules for addition in Val_semilattice. For instance, you could reformulate $\text{gamma}\_\text{plus}'$ as: $i1 \in \gamma a1 \implies i2 \in \gamma a2 \implies i = i1 + i2 \implies i \in \gamma(\text{plus}' a1 a2)$. (You will need to change the interface Val_semilattice).

You can start the formalization of the AI like this:

```plaintext
datatype bound = NegInf ("\infty^-") | PosInf ("\infty^+") | NaN | Real

datatype bounds = S "bound set"

instantiation bounds :: order
begin
  
definition less_eq_bounds where
    "x \leq y = (case (x, y) of (S x, S y) \Rightarrow x \subseteq y)"

  
definition less_bounds where
    "x < y = (case (x, y) of (S x, S y) \Rightarrow x < y)"
```