Abstract Interpretation: Exercises for day 2

February 3, 2015

1. Write a 3 counter machine program and interpret it using the operational semantics on an input of your choice

2. Systematically derive abstract operations for `=0`, `<>0`, `+1`, and `-1` on the Parity domain, starting from $\lambda a.\, \alpha \circ op \circ \gamma(a)$

   with $op \in \{\texttt{=0}, \texttt{<>0}, \texttt{+1}, \texttt{-1}\}$

   and where

$$
\begin{aligned}
- &: \wp(\mathbb{N}_0) \to \wp(\mathbb{N}_0) \\
\texttt{=0} &= \lambda S.\, \{s \mid s \in S \ \wedge \ s = 0\} \\
\texttt{<>0} &= \lambda S.\, \{s \mid s \in S \ \wedge \ s \neq 0\} \\
\texttt{+1} &= \lambda S.\, \{s + 1 \mid s \in S\} \\
\texttt{-1} &= \lambda S.\, \{s - 1 \mid s \in S \ \wedge \ s > 0\}
\end{aligned}
$$

   (note: $\mathbb{N}_0 = \{0, 1, 2, 3, \dots\}$)

3. Implement the Parity domain (its type, the constant $\bot$, and binary operations $\sqsubseteq$ and $\sqcup$) and the 4 above operations as an OCaml module with (roughly) the following signature:

```
type parity

val bot     : parity
val leq     : parity -> parity -> bool
val join    : parity -> parity -> parity

val iszero  : parity -> parity
val notzero : parity -> parity
val incr    : parity -> parity
val decr    : parity -> parity
```

4. Run your program from question 1 with the 3CM interpreter.
   Did you get the expected result?