Program Analysis, January–April 2011

Assignment 1, due 22 April 2011

> **Instructions for submitting solutions**
>
> *Please submit your solutions electronically by email to facilitate evaluation (I am travelling and will find it difficult to pick up and carry around physical submissions).*
>
> *Preferrably, send an electronic document in PDF (you can generate it in LaTeX or OpenOffice or Word or whatever, but send only PDF). If you can't do this, send scanned copies of handwritten pages.*

All exercises are taken from *Principles of Program Analysis* by Flemming Nielson, Hanne Riis Nielson and Chris Hankin. Cross check if you think there are typos!

1. **Exercise 2.3**

   A modification of the Available Expressions Analysis detects when an expression is available *in a particular variable*: a non-trivial expression a is available in x at a label ℓ if it has been evaluated and assigned to x on all paths leading to ℓ and if the values of x and the variables in the expression have not changed since then. Write down the data flow equations and any auxiliary functions for this analysis.

2. **Exercise 2.4**

   Consider the following program:

   $$[\texttt{x:=1}]^1;\ [\texttt{x:=x-1}]^2;[\texttt{x:=2}]^3$$

   Clearly `x` is dead at the exits from 2 and 3. But `x` is live at the exit of 1 even though its only use is to calculate a new value for a variable that turns out to be dead. We shall say that a variable is a *faint variable* if it is dead or if it is only used to calculate new values for faint variables; otherwise it is *strongly live.* In the example `x` is faint at the exits from 1, 2 and 3. Define a Data Flow Analysis that detects strongly live variables. (Hint: For an assignment $[x := a]^\ell$ the definition $f_\ell(l)$ should be by cases on whether x is in l or not.)

3. **Exercise 2.14**

   In a Detection of Signs Analysis one models all negative numbers by the symbol `-`, zero by the symbol `0` and all positive numbers by the symbol `+`. As an example, the set $\{-2, -1, 1\}$ is modelled by the set $\{\texttt{-},\texttt{+}\}$, that is an element of the powerset $\mathcal{P}(\{\texttt{-},\texttt{0},\texttt{+}\})$.

   Let S_* be a program and $\mathbf{Var}_*$ be the finite set of variables in S_*. Take L to be $\mathbf{Var}_* \to \mathcal{P}(\{\texttt{-},\texttt{0},\texttt{+}\})$ and define an instance $(L, \mathcal{F}, F, E, \iota, f.)$ of a Monotone Framework for performiNg Detection of Signs Analysis.

Similarly, take L' to be $\mathcal{P}(\mathbf{Var}_* \times \{\texttt{-},\texttt{0},\texttt{+}\})$ and define an instance $(L', \mathcal{F}', F', E', \iota', f'_\cdot)$ of a Monotone Framework for Detection of Signs Analysis. Is there any difference in the precision obtained by the two approaches?

4. **Exercise 2.15**

   In the previous exercise we defined a Detection of Signs Analysis that *could not* record the interdependencies between signs of variables (e.g. that two variables `x` and `y` always will have the same sign); this is sometimes called an *independent attribute analysis*. In this exercise we shall consider a variant of the analysis that is able to record the interdependencies between signs of variables; this is sometimes called a *relational analysis*. To do so take L to be $\mathcal{P}(\mathbf{Var}_* \to \{\texttt{-},\texttt{0},\texttt{+}\})$ and define an instance $(L, \mathcal{F}, F, E, \iota, f_\cdot)$ of a Monotone Framework for performing Detection of Signs Analysis. Construct an example showing that the result of this relational analysis may be more informative than that of the independent attribute analysis.