Homework 2: Overview of Program Analysis

Due: problems 1-5, January 30, 2006; problem 6, February 6, 2006; remaining problems, February 13, 2006.

In this homework we will get a bit more into the different kinds of program analysis.

If you wish, you can work in groups.

Sections 1.1-1.3: Reaching Definitions and Data Flow

Read sections 1.1-1.3 of our textbook: *Principles of Programming Analysis* by Flemming Nielson, Hanne Riis Nielson, and Chris Hankin (Springer-Verlag, 1999).

1. (10 points) Describe a question that an optimizing compiler might want to answer and could answer using information from the Reaching Definitions analysis. For example, a compiler might want to know whether any variables might be assigned in a statement; this information could be obtained from the set of reaching definitions from the exits of all final labels (see section 2.1 of the text) in a statement S, by testing whether the set $\{\ell \mid (x, \ell) \in \mathsf{RD}_{exit}(\mathit{final}(S)), \ell \neq ?\}$ is empty. This might be useful for scheduling memory writes, or for reordering statements.

   Your task is to give another example.

2. (10 points) Consider the again your answer to the previous problem. Let us call the question described in your previous answer Q. Give one example of a statement (program) S and an answer to question Q that is incorrect for S. Explain how such an incorrect answer for Q could only be derived from an unsafe answer to the Reaching Definitions analysis for your statement S.

   For example, if Q is the example described in the previous problem, consider the statement: $[\mathtt{y} := 641]^1$. If the answer to Q was, incorrectly, that no variables are assigned in that statement, it could only be from an unsafe answer to the Reaching Definitions analysis, since the set $\{\ell \mid (x, \ell) \in \mathsf{RD}_{exit}(\mathit{final}(S)), \ell \neq ?\}$ could only be empty if $\mathsf{RD}_{exit}(1) = (\mathtt{y}, ?)$, which is unsafe.

   Your task is to explain the same kind of reasoning for your answer to the previous question.

3. (suggested practice) Do exercise 1.1 in the text.

4. (15 points; extra credit) What if we added arrays to the language? Describe how that would change the Reaching Definitions and give an example.

5. (15 points; extra credit) What kinds of language changes would make the Reaching Definitions analysis less precise? describe the syntax and semantics of the language change briefly, and how it affects the Reaching Definitions analysis.

Section 1.4: Constraint Based Analysis

Read sections 1.4 of our textbook: *Principles of Programming Analysis* by Flemming Nielson, Hanne Riis Nielson, and Chris Hankin (Springer-Verlag, 1999).

6. (20 points) Do the first part of exercise 1.2, showing that the solution in section 1.4 is a solution to the constraints. (For 10 points extra credit, you can show that it's also a least solution.)

Section 1.5: Abstract Interpretation

Read section 1.5 of our textbook: *Principles of Programming Analysis* by Flemming Nielson, Hanne Riis Nielson, and Chris Hankin (Springer-Verlag, 1999).

7. (15 points) Do exercise 1.3 in the text. Note that α and γ are considered to be functions from sets to sets, as in section 1.5.

8. (40 points) Do exercise 1.4 in the text; however to avoid extra work, for the first part of this exercise, you only have to show that

$$\alpha(G_j(\gamma(\mathsf{RD}_1), \ldots, \gamma(\mathsf{RD}_{12}))) \subseteq F_j(\mathsf{RD}_1, \ldots, \mathsf{RD}_{12})$$

for $j = 4, 5, 6$, i.e., for $G_4 = G_{exit(2)}$, $G_5 = G_{entry(3)}$, and $G_6 = G_{exit(3)}$.

Section 1.6: Type and Effect Systems

Read section 1.6 of our textbook: *Principles of Programming Analysis* by Flemming Nielson, Hanne Riis Nielson, and Chris Hankin (Springer-Verlag, 1999).

9. (10 points) Using the type system specified in Table 1.2, what is the least type of the following program?

$$[\texttt{q := 0}]^1\texttt{; if } [\texttt{r > q}]^2 \texttt{ then } [\texttt{x := r}]^3 \texttt{ else } [\texttt{y := r}]^4$$

10. (10 points) Using the type system specified in Table 1.3, what is the least type of the following program?

$$[\texttt{q := 0}]^1\texttt{; if } [\texttt{r > q}]^2 \texttt{ then } [\texttt{x := r}]^3 \texttt{ else } [\texttt{y := r}]^4$$

Section 1.7: Algorithms

Read section 1.7 of our textbook: *Principles of Programming Analysis* by Flemming Nielson, Hanne Riis Nielson, and Chris Hankin (Springer-Verlag, 1999).

11. (suggested practice) Do exercise 1.7.

12. (suggested practice) Do exercise 1.8.

Section 1.8: Transformations

Read section 1.8 of our textbook: *Principles of Programming Analysis* by Flemming Nielson, Hanne Riis Nielson, and Chris Hankin (Springer-Verlag, 1999).

13. (suggested practice) Do exercise 1.9.

14. (15 points; extra credit) Do exercise 1.10.