CS5214 – THE DESIGN OF OPTIMISING COMPILERS

Assignment 3

Due: 25 March 2016

Notes for the assignment:

1. This assignment is to be done on an individual basis. While you can discuss the assignment with others as much as is necessary, plagiarism will be doubly penalized – both the copier *and* the copied will be penalized.

2. The assignment is due 25th March 2016 at 11.59pm. This is a hard deadline.

3. This assignment involves more pen-and-paper exercise than programming. Please submit a PDF with your name and matric number clearly stated in the name of the document as well as within the document in the IVLE folder that I have created for this assignment.

Description of Assignment 3

Part 1 – Loop analysis

Consider the following for loop:

```
for (i=1; i<11; i++)
   for (j=1; j<51; j++)
      for (k=1; k<21; k++)
         X[3*j+k, i+2*j+k, 2*j+3*k] = ... X[2*i+4*k, 5*j+7*k, 3*i+5*j] ...
```

(a) Formulate the integer problem for the dependence analysis between the two references to the array X.
(b) Using Fourier-Motzkin elimination, check if **real** solutions exist.
(c) Perform the Omega test and try to obtain all **integer** solutions.
(d) Give a C program that will exhaustively search for all solutions for this particular problem. Do the solutions agree with what you found in (c). If not, what is the problem and how do you overcome it?

Part 2 – Loop tiling

Consider the following simple loop:

```
int X[N,N], Y[N,N];
…
    for (i=0; i<N; i++)
        for (j=0; j<N; j++)
                X[i, j] = Y[j, i];
```

Show how tiling will help improve performance given a cache that can hold 8 integers in each cache line. In particular,

(a) Show how the upper limits of the new inner loops were derived, especially with respect to the original iteration space.
(b) Walk through the execution of the tiled loops to show how data in the cache lines are reused.
(c) Compute the total number of cache hits and misses before and after loop tiling.

Assume that the two dimensional array is laid out in row-major order and the array starts with the index 0.

_____ *END OF ASSIGNMENT 3* _____