Exercise Sheet 11

Compiler Design Lab
Prof. Dr. Sebastian Hack
Fabian Ritter, B.Sc.

Exercise 11.1 Schedules

In the following, we use a slightly generalized notion of schedules compared to what we have seen in the lectures. Here, a schedule $\Theta : \Omega \to \mathbb{Z}^k$ is a function mapping statement instances to vectors of integers. A statement instance ω_1 is executed before another statement instance ω_2 according to a schedule Θ iff $\Theta(\omega_1) \prec_{lex} \Theta(\omega_2)$, where $\prec_{lex}$ is the lexicographic extension of the usual < relation on integers.

1. Describe the difference between a statement and a statement instance and why this differentiation is useful.
2. Write down the schedule for a 2D loop nest that describes a loop interchange as a 2D matrix.
3. Write down the schedule for a 2D loop nest that describes a loop reversal of the inner loop as a 2D matrix.
4. How can the two schedules be combined? What does the combined scheduling matrix look like and what effect will it have on a 2D loop nest?
5. Describe when a schedule is considered legal with regards to a dependence relation Γ that contains all pairs of statement instances which depend on each other.

Exercise 11.2 Dependences

```
   for (int i = 0; i < N; i++)
     for (int j = 2; j < N; j++)
S:     C[j][i] = a * C[j-1][i] + b * C[j-2][i];
   for (int i = 0; i < N; i++)
P:   Out[i] = C[N-1][i];
```

1. Write down the iteration spaces (aka domain) of the statements S and P. Use a constraint system or visualize it in a 2D coordinate system.
2. Describe the dependences for the example above using dependence polyhedra. Characterize the dependences (RAW/WAR/WAW).
3. Describe a legal and beneficial transformation that could be applied and write down the transformed loop nest. Argue why the transformation is beneficial.
4. Which dimension in the example above (after your transformation) can be vectorized? What would the (pseudo) code look like?