HW 3: DATA DEPENDENCES

DIHAN DAI

February 22, 2020

Question 1

(a) The validity condition for loop unrolling is that the dependence vector of permutation corresponding to moving the unrolled loop to the innermost loop is valid. Therefore,
  - $(0, 0, 1, -1)$: unrolling t, i and k loops is valid.
  - $(1, -2, 1, -1)$: unrolling i, j and k loops is valid.
  - $(1, -1, 1, 0)$: unrolling i, j and k loops is valid.

(b) The validity condition for loop permutation is that the dependence vector of the corresponding permutation is valid.
  - $(0, 0, 1, -1)$: tijk, itjk, tjki, ijkt, jkti, jkit, tjik, ijtk, jtki, jikt, jitk, jtik
  - $(1, -2, 1, -1)$: tijk, tikj, tjki, tjik, tkij, tkji, jtik, jtki, jkit, jkti, jitk, jikt
  - $(1, -1, 1, 0)$: tijk, tikj, tjki, tjik, tkij, tkji, jtik, jtki, jkit, jkti, jitk, jikt, ktji, ktij, kjti, kjit

(c) The vaidity condition for loop tiling is that the corresponding loops are fully permutable.
  - $(0, 0, 1, -1)$: 3D tiling: tij, 2D tiling: ti, tj, ij,
  - $(1, -2, 1, -1)$: 3D tiling: ijk, 2D tiling: tj, ij, ik, jk,
  - $(1, -1, 1, 0)$: 3D tiling: ijk, 2D tiling: tj, ij, ik, jk,

Question 2

(a) Yes. If there is an output dependences between (t_1, i_1, j_1) and (t_2, i_2, j_2) such that $(t_1, i_1, j_1) < (t_2, i_2, j_2)$, it must satisfy $(i_1 + 1, j_1 - 1) = (i_2 + 1, j_2 - 1)$. The dependence vectors are given by $(t_2 - t_1, i_2 - i_1, j_2 - j_1)$ and the smallest of such vectors is $(1, 0, 0)$. The leftmost nonzero element of the vector is positive, thus the dependence vector is valid.

(b)
- Flow dependence: if there is a flow dependeces between the write instance (t_w, i_w, j_w) and the read instance(t_r, i_r, j_r). The dependences vector are given by $(t_r - t_w, i_r - i_w, j_r - j_w)$ and they must satisfies
  - either $(i_r, j_r + 1) = (i_w + 1, j_w - 1)$, where the smallest dependence vector is $(0, 1, -2)$,
  - or $(i_r - 1, j_r) = (i_w + 1, j_w - 1)$, where the smallest dependence vector is $(0, 2, -1)$.

  Both dependence vectors are valid because the leftmost nonzero elements of both vectors are positive.
- Anti-dependence: if there is an anti-dependeces between the read instance(t_r, i_r, j_r) and the write instance (t_w, i_w, j_w). The dependences vector are given by $(t_w - t_r, i_w - i_r, j_w - j_r)$ and they must satisfies
  - either $(i_r, j_r + 1) = (i_w + 1, j_w - 1)$, where the smallest dependence vector is $(1, -1, 2)$,
  - or $(i_r - 1, j_r) = (i_w + 1, j_w - 1)$, where the smallest dependence vector is $(1, -2, -1)$.

  Both dependence vectors are valid because the leftmost nonzero elements of both vectors are positive.

Question 3

(i)

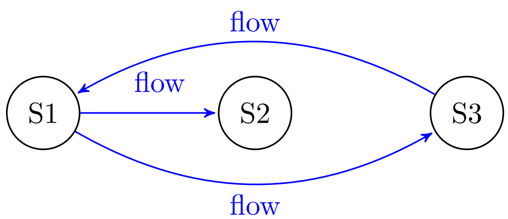
$$\text{i loop: } S1[i,j,k] \to S3[i+1,j,k],$$
$$\text{k loop: } S2[i,j,k] \to S1[i,j,k+1],$$
$$\text{j loop: } S3[i,j,k] \to S2[i,j+1,k].$$

(ii) No parallelization can be achieved since every loop carries dependence.

(iii) No. It is because the dependences of every loop doesn't change no matter what permutation of the nested loops is.

Question 4

(i) Dependence graph:



$$S1[i] \xrightarrow{\text{flow}} S2[i]$$
$$S1[i] \xrightarrow{\text{flow}} S3[i+1]$$
$$S3[i] \xrightarrow{\text{flow}} S1[i+1]$$

Statement 1 and 3 form a cycle, thus only statement 2 can be vectorized.

(ii)
```
for(i=1; i<256; i++)
{
    a[i] = c[i]+1;
    c[i+1] = a[i-1]+1;
}

b[1:255] = a[1:255]-1;
```

Question 5

(i) Data dependences:

$$S[i,j,k] \to S[i+1,j-1,k-1] \text{ flow}$$

(ii)
```
for(i=1; i<256; i++)
{
  a[i][1:255][1:255] = a[i-1][2:256][2:256]+1;
}
```