

Laboratorium 8. Zastosowanie tranzytywnego domknięcia do implementacji techniki blokowania pętli (patrz wykład 13), 2 godziny.

Zadania:

1. Dla wskazanej pętli za pomocą kalkulatora ISCC znaleźć relację zależności, R, oraz przestrzeń iteracji, LD.
2. Zrobić rysunek pokazujący zależności w przestrzeni 6 x 6. W tym celu trzeba zastosować operator `scan (R*[n]->{:n=6})`; który wygeneruje wszystkie zależności w przestrzeni 6 x 6, pierwsza krotka wskazuje początek zależności (strzałki), druga krotka – koniec zależności (strzałki).
!!!Uwaga: dla niektórych pętli w przestrzeni 6x6 zależności mogą nie istnieć, w takim przypadku należy rozszerzyć przestrzeń do rozmiaru 12x12.
3. Utworzyć zbiór TILE dla rozmiaru kafelka 2x2
4. Utworzyć zbiór TILE_LT
5. Utworzyć zbiór TILE_GT
6. Obliczyć relację R+
7. Obliczyć zbiór TILE_ITR
8. Obliczyć zbiór TVLD_LT
9. Obliczyć zbiór TILE_VLD
10. Sprawdzić zawartość kafelków za pomocą operatora `scan` kalkulatora `iscc`
11. Utworzyć zbiór TILE_VLD_EXT
12. Przekształcić zbiór TILE_VLD_EXT na relacje CODE
13. Wygenerować kod za pomocą operatora `codegen`
14. Zastosować program porównujący wyniki obliczeń do sprawdzania poprawności kodu docelowego w przestrzeni 6x6.
15. Opracować sprawozdanie.

Patrz skrypt „L8” pokazujący dla przykładowej pętli realizację poszczególnych kroków zadań wyżej.

Warianty pętli:

1.

```
for(i=4;i<=n;i++)  
  for(j=4;j<=n;j++)  
    a[i][j] = a[i][j-1];
```
2.

```
for(i=1;i<=n;i++)  
  for(j=2;j<=n;j++)  
    a[i][j] = a[i][j-2];
```
3.

```
for(i=1;i<=n;i++)  
  for(j=3;j<=n;j++)  
    a[i][j] = a[i][j-3];
```
4.

```
for(i=1;i<=n;i++)
```

```

    for(j=1;j<=n;j++)
        a[i][j] = a[i-1][j-1];
5.
for(i=2;i<=n;i++)
    for(j=1;j<=n;j++)
        a[i][j] = a[i-2][j-1];
6.
for(i=2;i<=n;i++)
    for(j=2;j<=n;j++)
        a[i][j] = a[i-2][j-2];
7.
for(i=2;i<=n;i++)
    for(j=2;j<=n;j++)
        a[i][j] = a[i-2][j+2];
8.
for(i=1;i<=n;i++)
    for(j=0;j<=n;j++)
        a[i][j] = a[i-1][j+2];
9.
for(i=1;i<=n;i++)
    for(j=0;j<=n;j++)
        a[i][j] = a[i-1][j+1];
10.
for(i=1;i<=n;i++)
    for(j=0;j<=n;j++)
        a[i][j] = a[i+3][j+4];
11.
for(i=1;i<=n;i++)
    for(j=4;j<=n;j++)
        a[i][j] = a[i+3][j-4];
12.
for(i=1;i<=n;i++)
    for(j=4;j<=n;j++)
        a[i][j] = a[i+4][j-4];
13.
for(i=1;i<=n;i++)
    for(j=4;j<=n;j++)
        a[i][j] = a[i+5][j-4];

```

Sprawozdanie powinno zawierać: pętlę, skrypt implementujący zadania oraz wyniki wszystkich zadań.