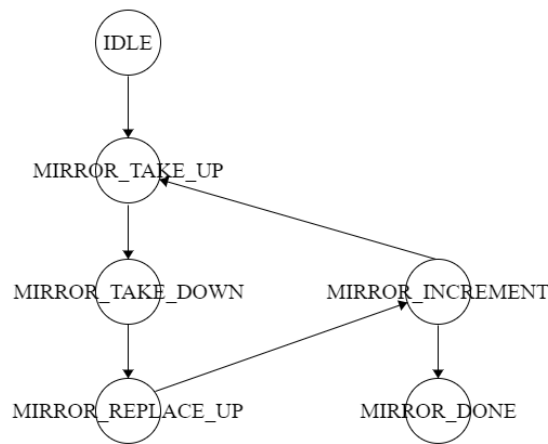


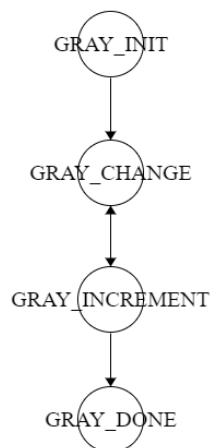
## Tema 1 - Prelucrarea imaginilor

### 1. Oglindirea:



Pentru oglindire am ales să parcurg jumătatea superioară a imaginii, interschimbând elementele de pe linia  $k$  cu cele de pe linia  $63-k$ . Pentru asta, am stocat pixelii în două variabile auxiliare. În **MIRROR\_TAKE\_UP**, primul pixel (cel de pe rândul de sus) este memorat. În **MIRROR\_TAKE\_DOWN**, pixelul de jos este memorat și schimbat, după care revenim pentru a modifica valoarea pixelului de pe rândul de sus. În cele din urmă, se incrementează coloana (și dacă este nevoie și rândul) sau, dacă am ajuns la jumătatea imaginii, se marchează procesul ca fiind încheiat.

### 2. Trecerea în alb-negru

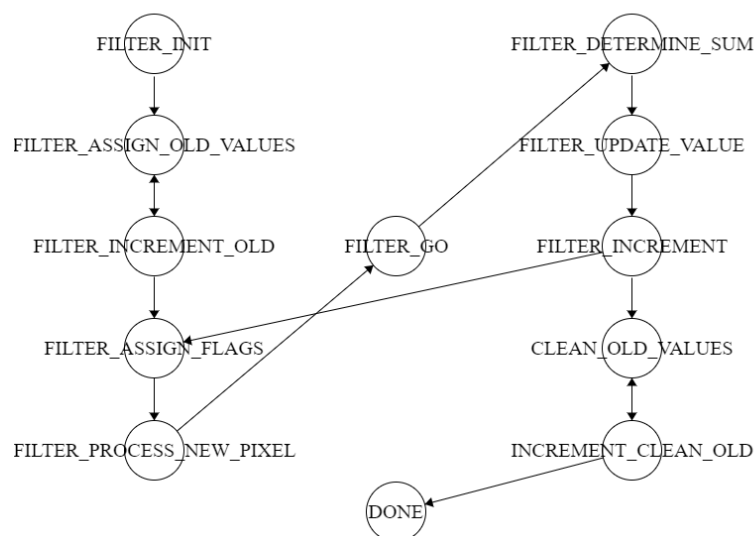


Media valorilor minime și maxime a fost calculată folosind formula:

$$((R < G) ? ((R < B) ? R : B) : ((G < B) ? G : B)) + ((R > G) ? ((R > B) ? R : B) : ((G > B) ? G : B)) / 2;$$

Prima parte (înainte de „+”) reprezintă minimul, iar a doua maximul. Pentru lizibilitate, am definit R, G și B (folosind directiva *define*) ca fiind porțiunile corespunzătoare din registrul in\_pix. Analog, R\_o, G\_o, B\_o pentru out\_pix.

### 3. Filtrul de sharpness:



Ideea centrală constă în salvarea valorilor vechi pe poziția elementului roșu din pixeli, pentru a le putea folosi după schimbarea valorilor principale, cele de pe canalul verde.

Procesarea fiecărui pixel începe cu setarea a 8 flag-uri, fiecare reprezentând existența în imagine a unui element învecinat. De exemplu, pentru elementele de pe primul rând, ul (upper left), um (upper mid) și ur (upper right) vor fi setate ca fiind 0. Pentru fiecare flag setat la 1, elementul corespunzător va fi scăzut din sumă.

**FILTER\_GO** reprezintă, de fapt, 8 stări (l-am reprezentat în diagramă ca fiind una singură pentru simplitate). În fiecare stare este preluat unul dintre elementele învecinate. Aici se realizează citirea și scăderea pentru elementele cu flag-ul 1.

În cele din urmă, valorile de pe canalul roșu sunt setate înapoi la 0.