Súlyozott automaták alkalmazása képek reprezentációjára

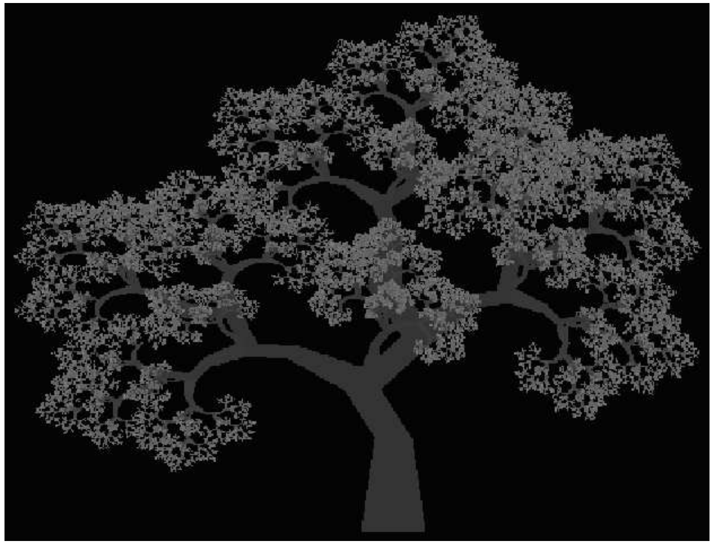
Gazdag Zsolt

Szegedi Tudományegyetem
Számítástudomány Alapjai Tanszék

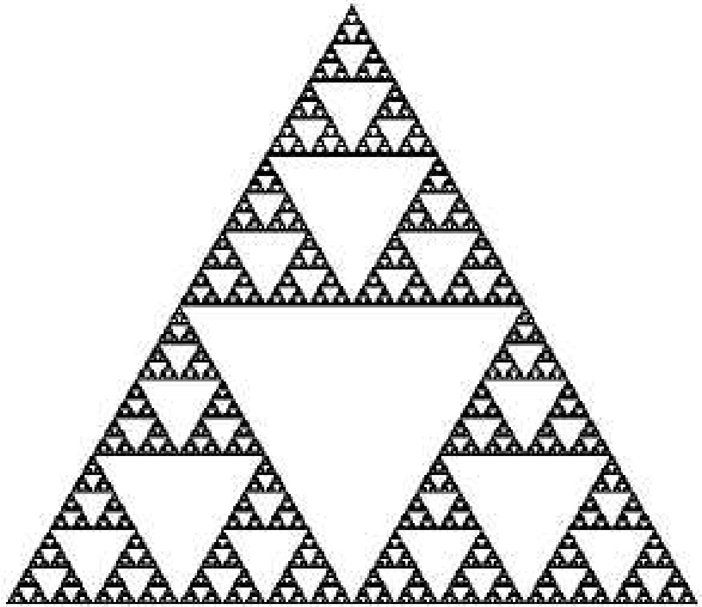
- Motiváció
- Fraktáltömörítés
- Súlyozott véges automaták
- Képek reprezentációja sv-automatákkal
- Példák

- A digitális kép általában igen nagy viszont redundás adathalmaz (512*512*8 bit $\approx$ 2 Mbyte)
- Tömörítéssel csökkenthető a kép mérete
  - Veszteségmentes tömörítés (futáshossz kódolás, Huffman kódolás)
  - Veszteséges tömörítés (DCT, <u>fraktáltömörítés</u>)

Fraktálok

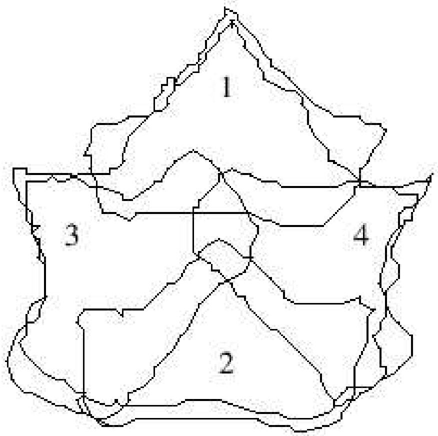


Egyszerű fraktál fa



Sierenpinski háromszög

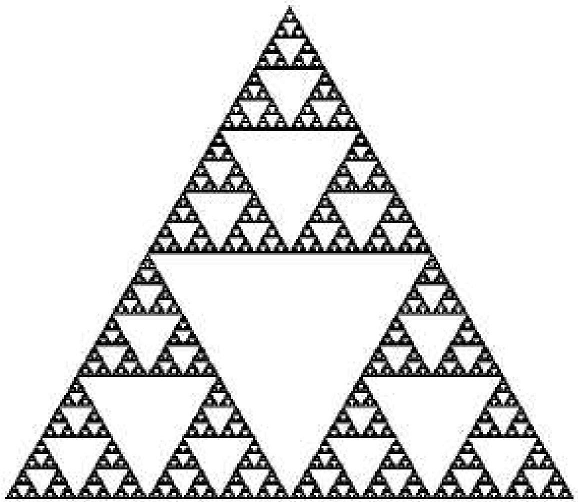
IFS - Iterált függvényrendszer (M. Barnsley)



Egy levél közelítése önmaga transzformált másolataival

A fenti transzformációk által meghatározott IFS határértéke

IFS - Iterált függvényrendszer (M. Barnsley)



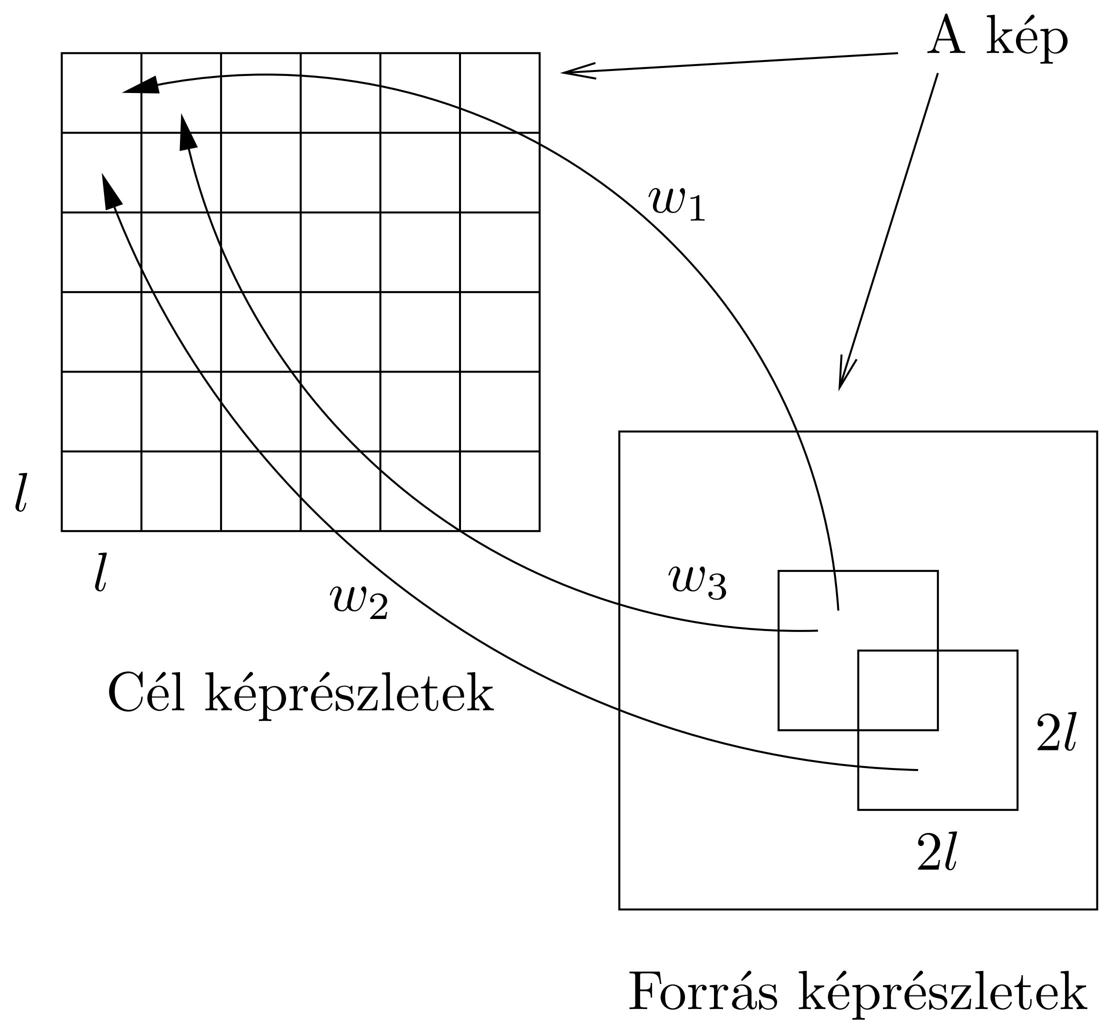
$$w_1 = \begin{bmatrix} \frac{1}{2} & 0 \\ 0 & \frac{1}{2} \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

$$w_2 = \begin{bmatrix} \frac{1}{2} & 0 \\ 0 & \frac{1}{2} \end{bmatrix} + \begin{bmatrix} \frac{1}{2} \\ 0 \end{bmatrix}$$

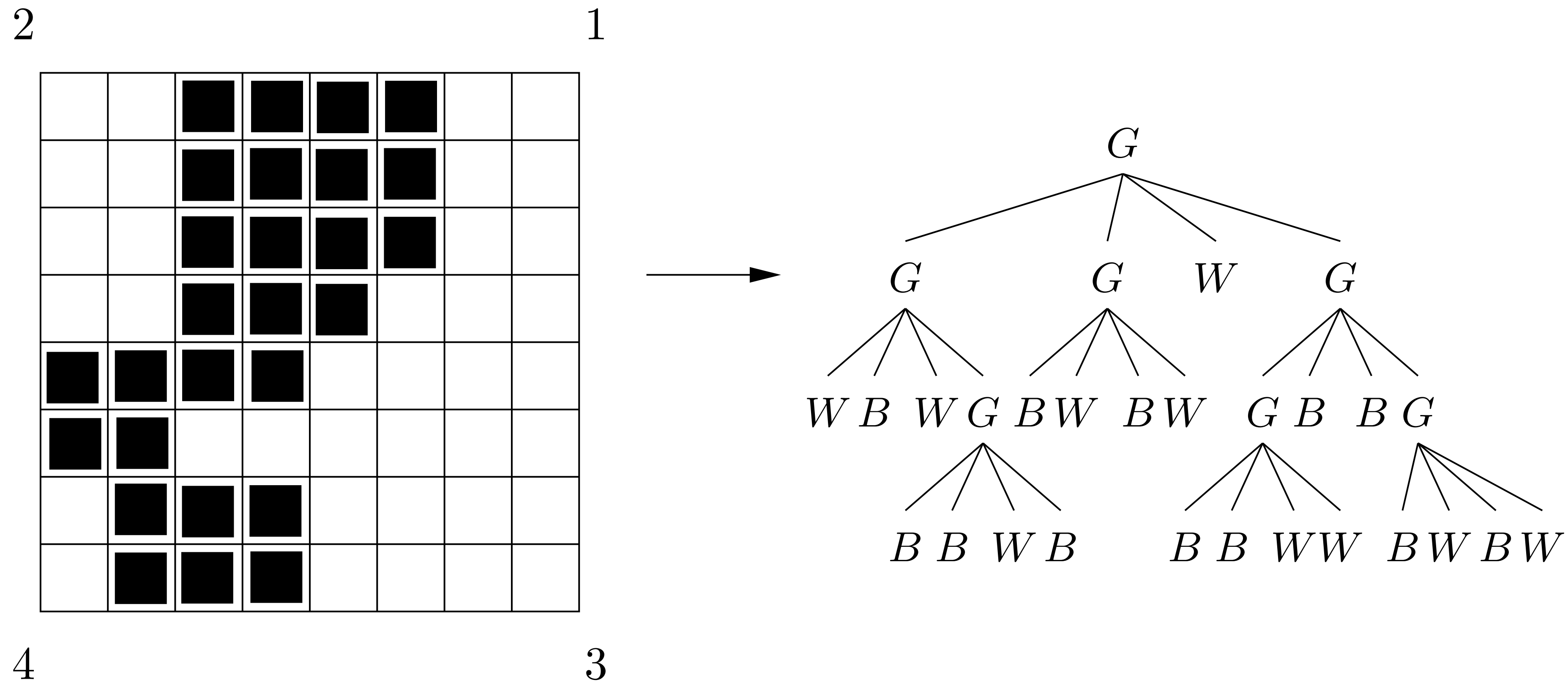
$$w_3 = \begin{bmatrix} \frac{1}{2} & 0 \\ 0 & \frac{1}{2} \end{bmatrix} + \begin{bmatrix} \frac{1}{4} \\ \frac{1}{2} \end{bmatrix}$$

A Sierenpinski háromszög és az őt indukáló iterált függvényrendszer

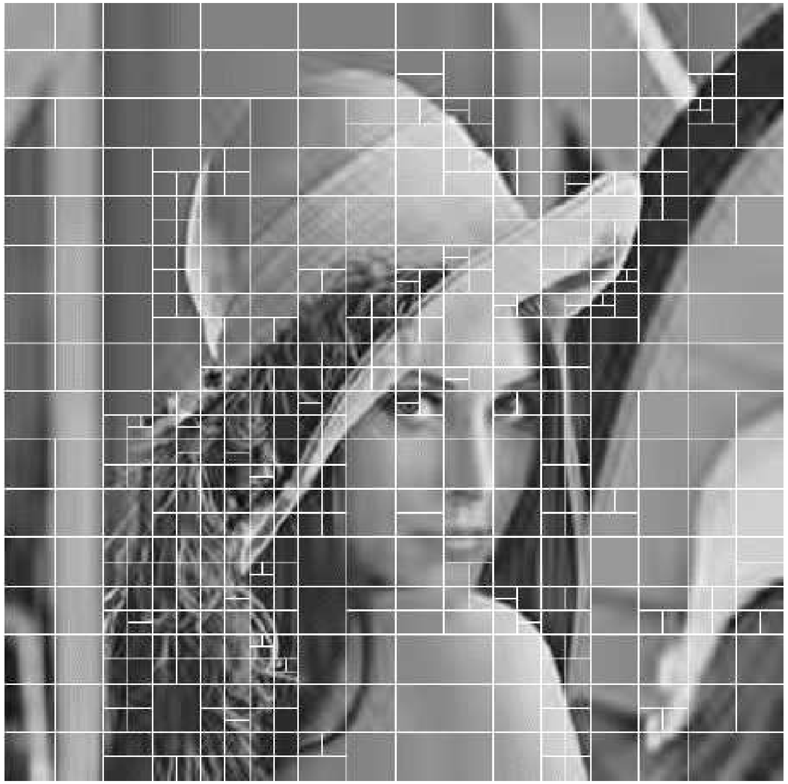
PIFS - Partícionált Iterált függvényrendszer (A. Jacquin)



Négyágú fák (quadtree coding)



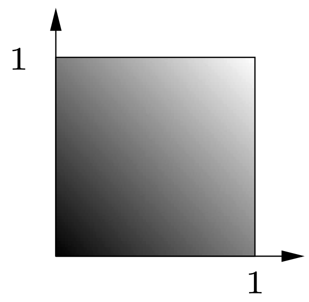
A kódolandó kép és az őt kódoló négyágú fa



Lena partícionálása a bináris fa kódolással

Definíciók

- Végtelen felbontású kép: $[0,1] \times [0,1] \to \mathbb{R}$

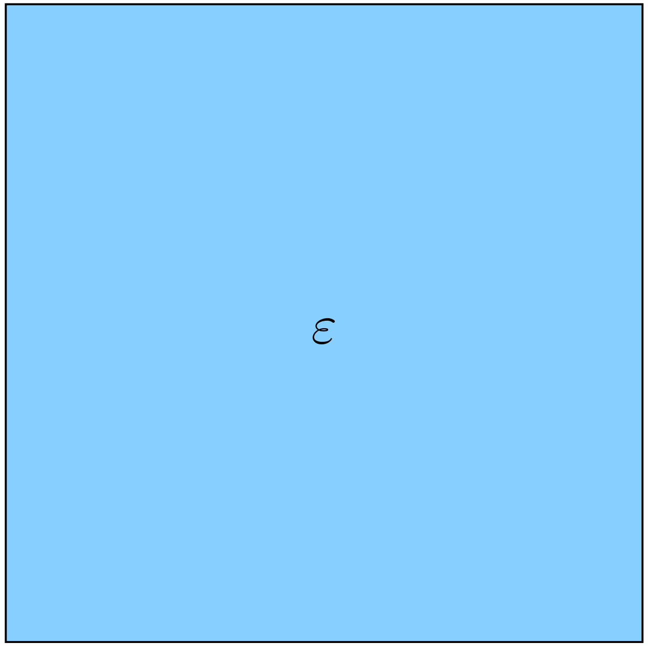


$z = x + y$

- Σ feletti legfeljebb n hosszú szavak: Σ^n

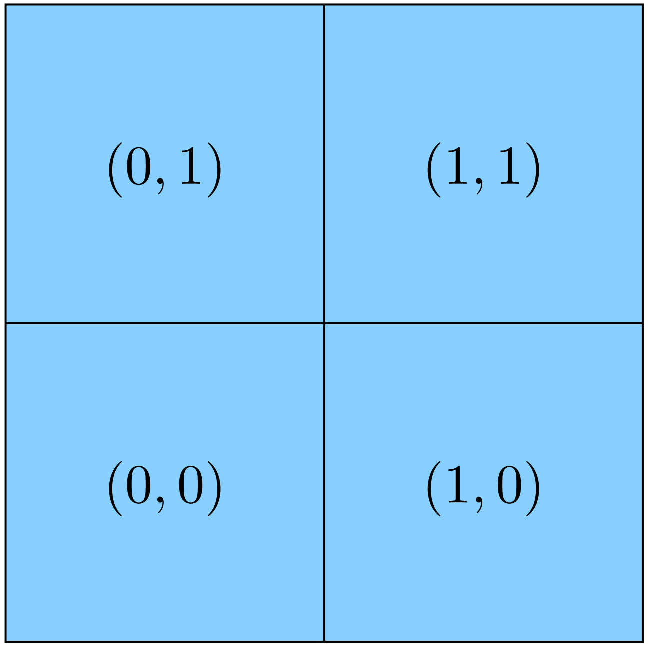
- Σ feletti véges szavak: Σ^*

Képrészletek címkézése



Egy kép szegmenseinek címkézése

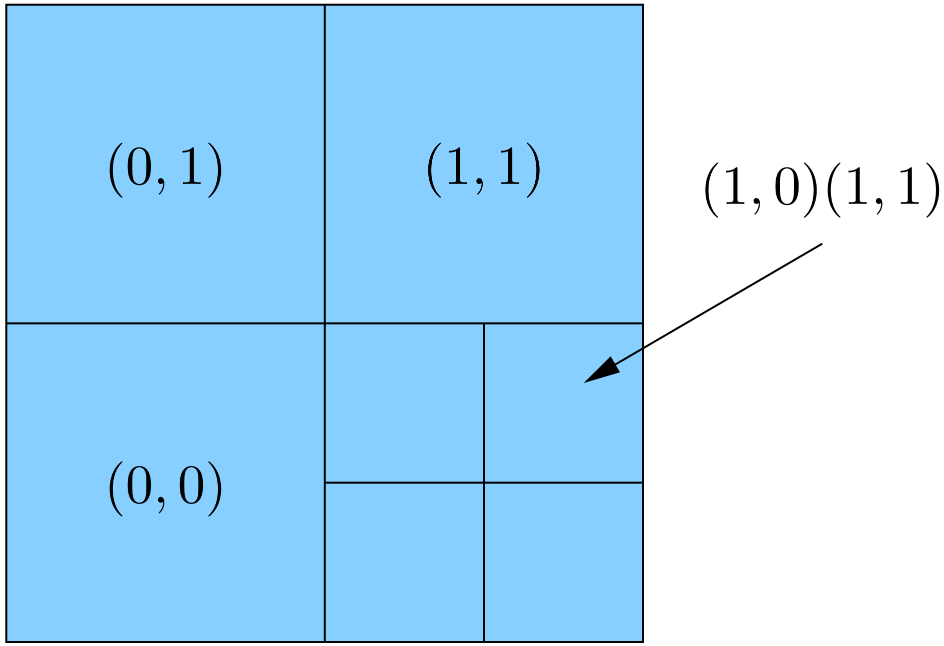
$\Sigma = \{(0,0),(0,1),(1,0),(1,1)\}$ feletti szavakkal



Egy kép szegmenseinek címkézése

$\Sigma = \{(0,0),(0,1),(1,0),(1,1)\}$ feletti szavakkal

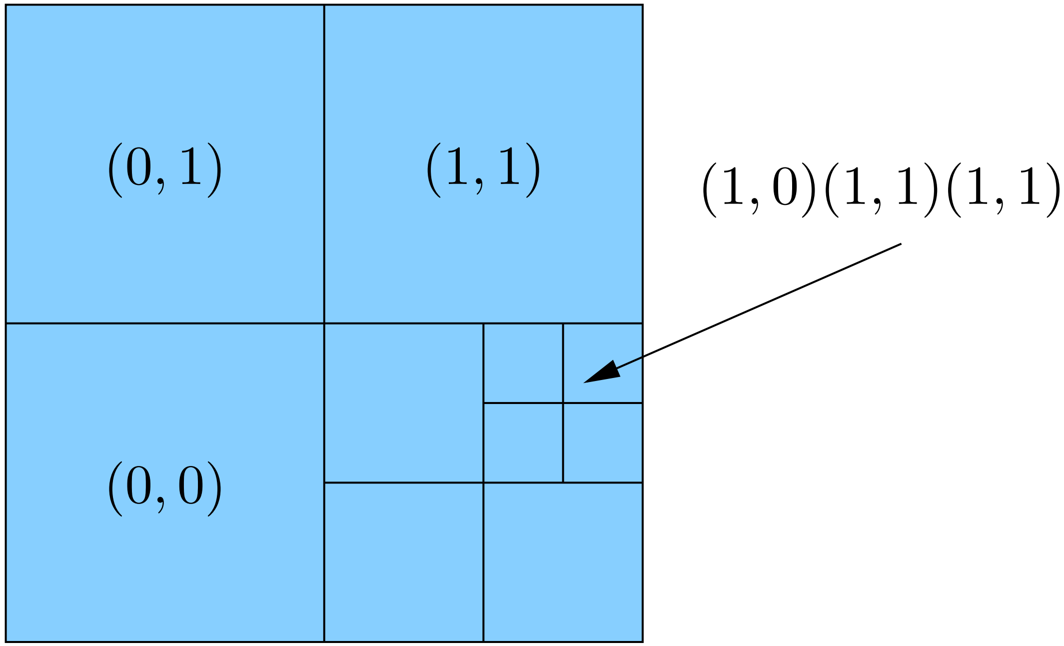
Képrészletek címkézése



Egy kép szegmenseinek címkézése

$\Sigma = \{(0,0),(0,1),(1,0),(1,1)\}$ feletti szavakkal

Képrészletek címkézése



Egy kép szegmenseinek címkézése

$\Sigma = \{(0,0),(0,1),(1,0),(1,1)\}$ feletti szavakkal

A továbbiakban $\Sigma = \{(0,0),(0,1),(1,0),(1,1)\}$

- Többszörös felbontású kép: $f : \Sigma^* \rightarrow \mathbb{R}$

  (Minden végtelen felbontású képhez megadható egy őt reprezentáló többszörös felbontású kép)

- Ha $f_1, f_2 : \Sigma^* \rightarrow \mathbb{R}$ és $c \in \mathbb{R}$, akkor minden $w \in \Sigma^*$:

$$(f_1 + f_2)(w) = f_1(w) + f_2(w), \quad (cf_1)(w) = cf_1(w)$$

- f átlagmegőrző: $\forall w \in \Sigma^* : \quad f(w) = \frac{1}{4}\Sigma_{a\in\Sigma} f(wa)$

A továbbiakban csak átlagmegőrző fügvényekkel foglalkozunk!

Súlyozott véges automata (sv-automata):
$\mathcal{A} = (Q, \Sigma, W, I, F)$

- Q: véges állapothalmaz ($Q = \{q_1, \ldots, q_k\}$)
- Σ: input ábécé
- $W : Q \times \Sigma \times Q \to \mathbb{R}$, a súlyfüggvény ($(W_a)_{i,j} = W(i, a, j)$)
- $I, F : Q \to \mathbb{R}$, a kezdő illetve befejező hozzárendelések (oszlopmátrixok)

Az $\mathcal{A}$ által q_j-ben indukált többszörös felbontású kép:

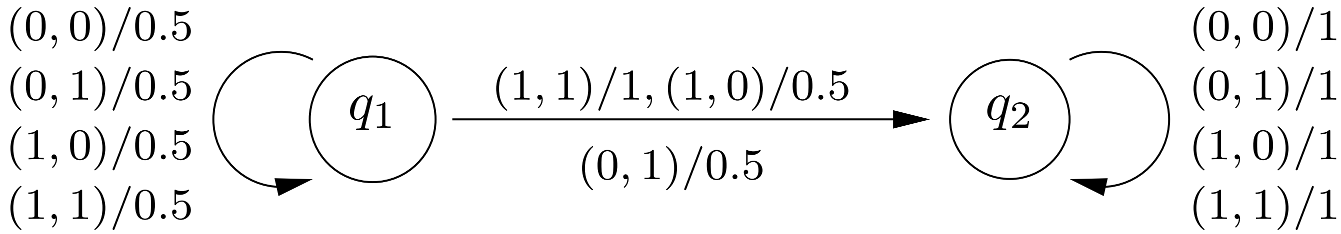
$$f_j(a_1 a_2 \dots a_n) = e_j^t W_{a_1} \dots W_{a_n} F$$

Az $\mathcal{A}$ által indukált többszörös felbontású kép:

$$f_{\mathcal{A}} = \sum_{j=1}^{k} I_j f_j$$

$\mathcal{A}$ átlagmegőrző, ha $f_{\mathcal{A}}$ átlagmegőrző

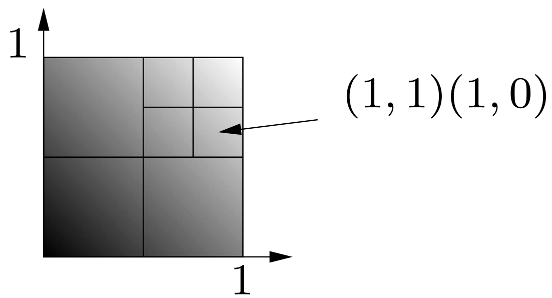
Súlyozott véges automata – Példa



$$I = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \; F = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$$

$$f_{\mathcal{A}}((1,1)(1,0)) = [1,0] \begin{bmatrix} 0.5 & 1 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 0.5 & 0.5 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 \\ 1 \end{bmatrix} = 1.5$$

$f_{\mathcal{A}}$ konvergál a $z = x + y$ függvényhez:



Az első sv-automata kódoló algoritmus (Culik és Kari)

Input: Egy f átlagmegőrző függvény

Output: Egy $\mathcal{A}$ sv-automata, amire $f_{\mathcal{A}} = f$

Hozzunk létre egy állapotot amely az egész képet reprezentálja;

while (vannak feldolgozatlan állapotok)

{

Vállasszunk egy feldolgozatlan állapotot;

Az állapotban indukált kép minden negyedére:

{

if (az aktuális negyed már létező állapotok
által indukált képek lineáris kombinációja)

Tároljuk az együtthatókat;

else

{

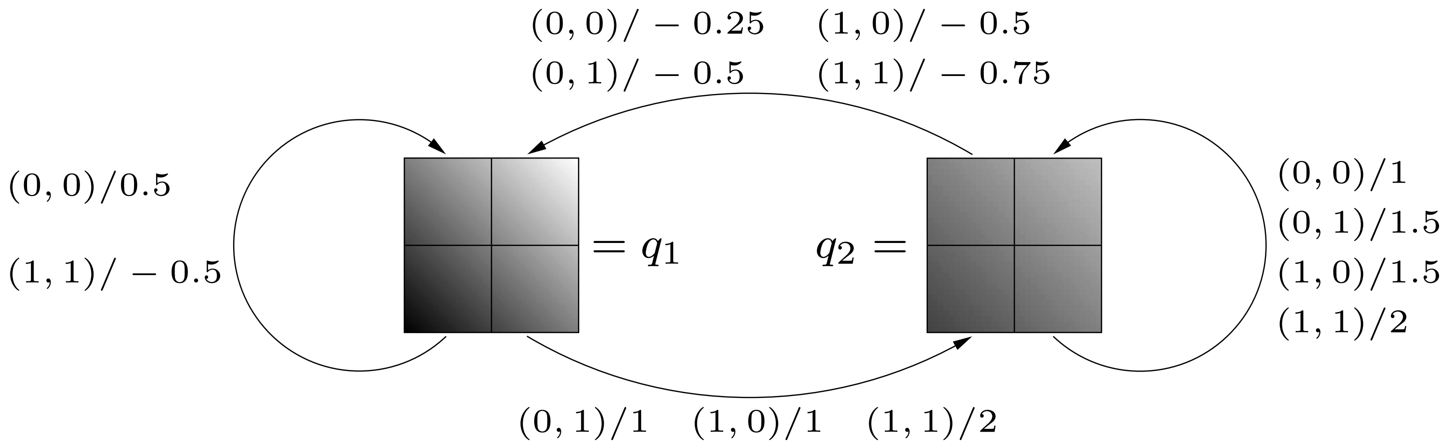
Hozzunk létre egy az aktuális képnegyednek
megfelelő új állapotot;

Vegyük fel ezt az állapotot a feldolgozandó
állapotok listájába;

}}}

Legyen $\mathcal{A}$ a kapott sv-automata

Ha f a $z = x + y$ függvényhez konvergáló átlagmegőrző függvény, akkor az output automata:



$$I = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \; F = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$$

Például: $f((1,1)) = 2f_2 - 0.5f_1$

Tétel. Az $\mathcal{A}$ sv-automata minimális állapotszámú a f-t kiszámló sv-automaták között [CK93].

Probléma: Hogyan lehet csökkenteni $\mathcal{A}$ átmeneteinek számát és az algoritmus futási idejét?

- Csak közelítjük a képeket (megengedünk egy hibát; kis súlyú éleket elhagyjuk)
- Megnézzük, hogy mennyire jó képet kapunk, ha közelítünk egy szegmenst illetve, ha tovább osztjuk. Ha kell visszalépünk (backtrack) [CK94]

AutoPic (F. Katritzke, W. Merzenich, M. Thomas)

A tömörített Lena (PSNR: 32.89 dB, 0.2246 bpp, 880 állapot)

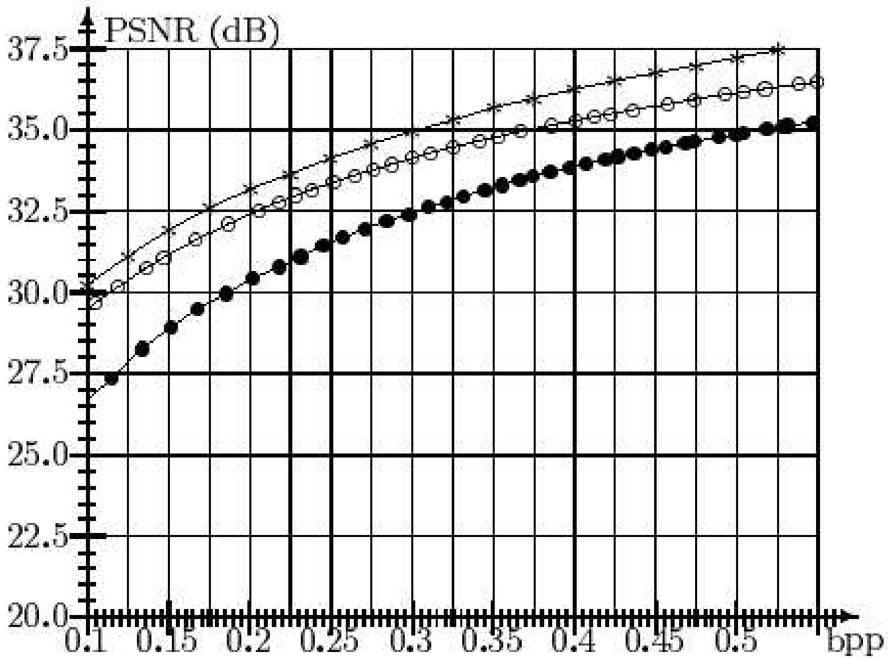
A tömörített Boat (PSNR: 31.85 dB, 0.3344 bpp, 1276 állapot)

Futási idő (Lena, 500 MHz Pentium III, 128 Mb RAM)

0.2057 bpp, tömörítés: 60.1 mp, kicsomagolás: 0.84 mp

0.2468 bpp, tömörítés: 65.5 mp, kicsomagolás: 0.97 mp

AutoPic (F. Katritzke, W. Merzenich, M. Thomas)



○ : AutoPic, • : IJPEG *: SPIHT (wavelet)

Lena tömörítése három különböző tömörítő programmal

Irodalom

M.F. Barnsley. Fractal Everywhere. Academic Press, New York, 1988.

K. Culik II és J. Kari. Image compression using weighted finite automata. In *MFCS*, pages 392–402, 1993.

K. Culik II és J. Kari. Image-data compression using edge-optimizing algorithm for wfa inference. *Inf. Process. Manage.*, 30(6):829–838, 1994.

A. E. Jacquin. Image coding based on a fractal theory of iterated contractive image transformations. IEEE Trans. Image Processing, vol. 1, pp. 1830, 1992.

F. Katritzke. Refinements of data compression using weighted finite automata.

Ph.D. Thesis, University of Siegen, 2001.