# Váhované konečné automaty a jejich aplikace ve zpracování obrazu

## TID

Autor: Filip Kadlček
ikadlcek@fit.vutbr.cz

Vedoucí: doc. Dr. Ing. Otto Fučík
fucik@fit.vutbr.cz

# Obsah

- Konečný automat a konečný převodník
- WFA a WFT
- Adresování pixelů v obraze
- Černobílý obraz
- Obraz ve stupních šedi
- Aplikace WFA a WFT ve zpracování obrazu

# Motivace

- Získání specifikace obrazu (šedotónového), jako (lokální) funkci nad stupni šedi
  - Popis obrazů pomocí formálních prostředků
  - Nové metody pro kompresi obrazu
  - Formální popis fraktálů a tvorba obrázků s nekonečným rozlišením

# Konečný automat

(Nedeterministickým) Konečným automatem rozumíme jednocestný iniciální automat M, který je specifikován 5-ticí :

$$M=(Q,\Sigma,\delta,q_0,F)$$

kde:

1. Q je konečná množina stavů
2. Σ je konečná vstupní abeceda
3. δ je funkce přechodů tvaru : $\delta : Q \times \Sigma \rightarrow 2^Q$

   $$\delta \subset Q \times \Sigma \times Q$$

4. $q_0 \in Q$ je počáteční stav
5. $F \subseteq Q$ je množina koncových stavů

# Konečný převodník

(Nedeterministickým) Konečným převodníkem T je stroj specifikován 6-ticí :

$$M=(Q,\Sigma,\Gamma,\delta,q_0,F)$$

kde:

1. Q je konečná množina stavů
2. Σ je konečná vstupní abeceda
3. Γ je konečná výstupní abeceda
4. δ je funkce přechodů tvaru : $\delta : Q \times (\Sigma \cup \{\varepsilon\}) \rightarrow 2^{Q \times (\Gamma \cup \{\varepsilon\})}$

$$\delta \subseteq Q \times (\Sigma \cup \{\varepsilon\}) \times (\Gamma \cup \{\varepsilon\}) \times Q$$

1. $q_0 \in Q$ je počáteční stav
2. $F \subseteq Q$ je množina koncových stavů

Mealyho automat

# Váhovaný konečný automat - WFA

- Rozšíření konečných automatů
- Přiřazení vah ke každému přechodu
- Přiřazení vah ke každému stavu (počáteční a koncové váhy)
- Výsledkem je neočekávaná výpočetní síla
- Vhodné pro zpracování obrazu

# WFA - definice

Váhovaný konečný automat A je popsán pomocí n-tice

$$A = (Q,\Sigma,i,t,w)$$

Kde:

1. Q je konečná množina stavů
2. Σ je konečná vstupní abeceda
3. i je počáteční ohodnocení stavů, definováno jako funkce $i : Q \rightarrow \Re$
4. t je koncové ohodnocení stavů, definováno jako funkce $t : Q \rightarrow \Re$
5. w je ohodnocení přechodů mezi stavy, definováno jako funkce

$$w : Q \times \Sigma \times Q \rightarrow \Re$$

# Deterministický WFA

- WFA **A** je **deterministický**, pokud zanedbáním vah vznikne deterministický konečný automat
- tj. pokud pro každý stav $p \in Q$ a $a \in \Sigma$ dostaneme jen jeden stav $q \in Q$, pro který platí :

$$w(p,a,q) \neq 0$$

- **Deterministický WFA je slabší než nedeterministický WFA**
  - Deterministické average preserving-WFA mohou generovat jenom spočetnou množinu fraktálů nebo funkci konstantní úrovně šedi
  - Nemůže generovat spojitě rostoucí funkci šedi

# WFA - výpočet

Pro každý WFA A přiřazujeme **distribuční funkci** $\delta_A : \mathrm{Q} \times \Sigma^* \to \Re$

$$\delta_A(p,\varepsilon) = t(p)$$

$$\delta_A(p,au) = \sum_{q \in Q} w(p,a,q) \cdot \delta_A(q,u) \quad \text{pro každé } p \in \mathrm{Q}, a \in \Sigma, u \in \Sigma^*$$

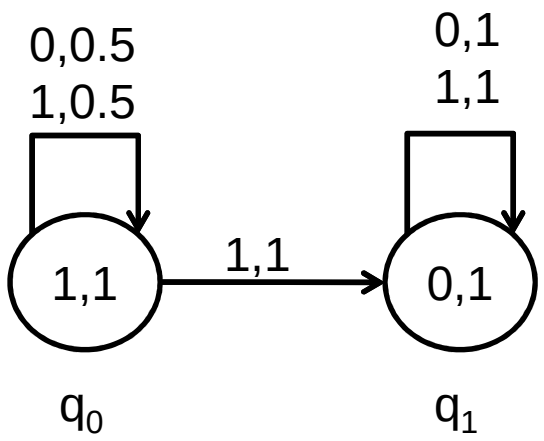
**Výpočetní funkce** WFA A je definována jako $f_A : \Sigma^* \to \Re$

$$f_A(u) = \sum_{p \in Q} i(p)\delta_A(p,u) \quad \text{pro každé } u \in \Sigma^*$$

# WFA - příklad

Mějme WFA A:

$$A = \left( \{q_0, q_1\}, \{0,1\}, (1,0), (1,1), \left\{ \begin{array}{l} w(q_0,1,q_0) = 0.5, w(q_0,0,q_0) = 0.5, \\ w(q_0,1,q_1) = 1, w(q_1,1,q_1) = 1, w(q_1,0,q_1) = 1 \end{array} \right\} \right)$$




$$\delta_A(q_0,011) = 0.5 \cdot 0.5 \cdot 0.5 \cdot 1 + 0.5 \cdot 0.5 \cdot 1 \cdot 1 + 0.5 \cdot 1 \cdot 1 \cdot 1 = 0.875$$

$$\delta_A(q_1,011) = 1 \cdot 1 \cdot 1 \cdot 1 = 1$$

$$f_A(011) = i(q_0) \cdot \delta_A(q_0,011) + i(q_1) \cdot \delta_A(q_1,011) = 1 \cdot 0.875 + 0 \cdot 1 = 0.875$$

# WFA – average preserving

Automat WFA A nazveme average preserving (průměrně zachovávající), pokud splňuje podmínku:

$$f_A(\omega) = \frac{1}{n} \sum_{a \in \Sigma} f_A(\omega a)$$

Nad abecedou $\Sigma$ a $n = |\Sigma|$, $\omega \in \Sigma^*$

# WFT - definice

Váhovaný konečný převodník T je popsán pomocí 6-tice

$$T = (\mathrm{Q}, \Sigma, \Gamma, i, t, w)$$

Kde:

1. Q je konečná množina stavů
2. Σ je konečná vstupní abeceda
3. Γ je konečná výstupní abeceda
4. *i* je počáteční ohodnocení stavů, definováno jako funkce $i : \mathrm{Q} \rightarrow \Re$
5. *t* je koncové ohodnocení stavů, definováno jako funkce $t : \mathrm{Q} \rightarrow \Re$
6. *w* je ohodnocení přechodů mezi stavy, definováno jako funkce

$$w : \mathrm{Q} \times (\Sigma \cup \{\varepsilon\}) \times (\Gamma \cup \{\varepsilon\}) \times \mathrm{Q} \rightarrow \Re$$

Automat je tzv. $\varepsilon$ – free jestliže je funkce ohodnocení přechodů ve tvaru

$$w : \mathrm{Q} \times \Sigma \times \Gamma \times \mathrm{Q} \rightarrow \Re$$

# WFT – váhovaná relace, vyčíslení WFT

WFT T určuje **váhovanou relaci** $R_T : \Sigma^* \times \Gamma^* \to \Re$

$$R_T(u,v) = \sum_{p,q \in Q} i(p) \cdot A_{p,q}(u,v) \cdot t(q)$$

Kde $p,q \in Q,\ u \in \Sigma^*, v \in \Gamma^*, A_{p,q}(u,v)$ je suma vah všech cesta začínajících ve stavu $p = p_1$ a končících ve stavu $q = p_{n+1}$ a jednotlivé hrany jsou popsány jako $u = a_1 \ldots a_n$ a $v = b_1 \ldots b_n$ *tj suma* $(p_1, a_1, b_1, p_2)\ (p_1, a_2, b_2, p_2) \ldots (p_n, a_n, b_n, p_{n+1})$

$A_{p,q}(u,v)$ může být někdy nekonečné, poté je celá relace nedefinovaná. Zdrojem nekonečnosti bývají epsilon přechody (automat poté může neustále cyklit a nezpracovávat vstup).

# Abeceda pro adresování pixelů

- Pomocí formálních jazyků můžeme reprezentovat jenom čtvercové obrázky, tj. obrazy s rozlišením $2^n$ x $2^n$ pixelů
  - Obrázky jiného rozlišení jsou doplněny na čtvercovou oblast a poté zpětně oříznuty
- Obraz popisujeme pomocí abecedy o čtyřech prvcích $\Sigma = \{0,1,2,3\}$
- Adresa každého pixelu obrazu, o rozlišení $2^n$ x $2^n$, se poté vyjádří jako řetězec nad danou abecedou o délce n, tj. řetězec nad $\Sigma^n$ např. pro $n = 6$ dostaneme „*021023*“
- Každý pixel obrazu tak představuje políčko o velikosti $4^{-n}$ z celkového obrazu
- Každý kvadrant (subkvadrant) je adresován pomocí jedné číslice
- ε adresuje celý obrázek

| | |
|---|---|
| 1 | 3 |
| 0 | 2 |

n = 1

| | | | |
|---|---|---|---|
| 11 | 13 | 31 | 33 |
| 10 | 12 | 30 | 32 |
| 01 | 03 | 21 | 23 |
| 00 | 02 | 20 | 22 |

n = 2

# Černobílý obraz

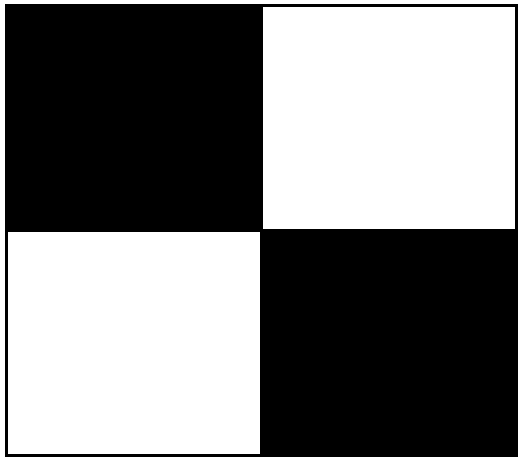
- Černobílý obraz o rozlišení $2^m$ x $2^m$ můžeme specifikovat pomocí Boolean funkce

$$\Sigma^m \rightarrow \{0,1\}$$

- Nebo jako množinu pixelů, které jsou černé, tj. jazyk $L \subseteq \Sigma^m$

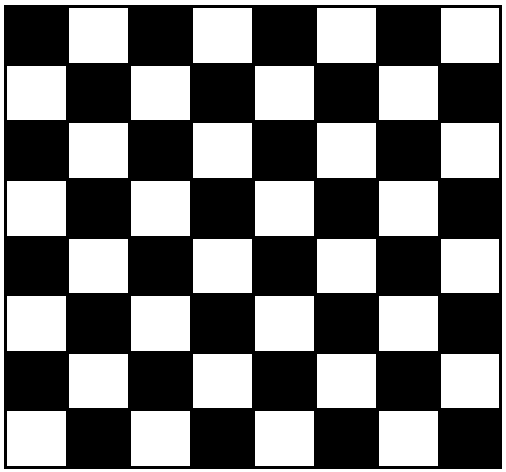
- **Multiresolution obraz**:
  - Má nekonečně mnoho stupňů rozlišení
  - Obraz s větším rozlišením má více detailů než obraz s menším rozlišením
  - Lze vyjádřit jako jazyk $L \subseteq \Sigma^*$, kde $\Sigma = \{0,1,2,3\}$, tj. množina bodů s černou bravou

# ČB obraz a KA - příklad



$$L_{22chess} = \{1,2\}\Sigma^*$$

- ve všech rozlišeních vypadá stejně ($2^n$ x $2^n$, pro n > 0)
- tj. jako obrázek popsaný jazykem *L = {1,2},*
- tj. obrázek v rozlišení *2 x 2* pixelů



$$L_{88chess} = \Sigma^2\{1,2\}\Sigma^* = \Sigma^2.L_{22chess}$$

- Opět ve všech rozlišení stejný ($2^n$ *x* $2^n$*, pro n > 2*)
- Vznikl konkatenací dvou jazyků
- Nahrazením jazyka $\Sigma^2$ za $\Sigma^n$, pro libovolné *n* dostaneme v konkatenaci s $L_{22chess}$ všechny možné obrázky s šachovnicemi

# ČB obraz, KA a fraktály- příklad

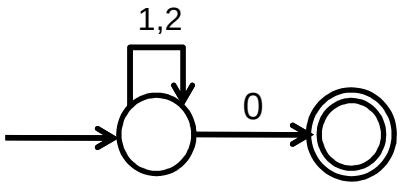
## Fraktál:

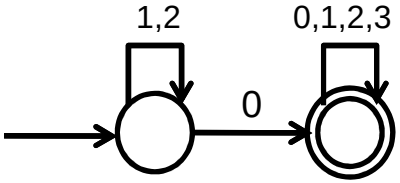
- Je obrazec, který když pozorujeme v jakémkoliv měřítku, tak se stále opakuje.
- Většinou se jedná o složitý obrazec, který je však popsán jednoduchou matematickou funkcí
- U fraktálů se předpokládá, že mají nekonečné rozlišení

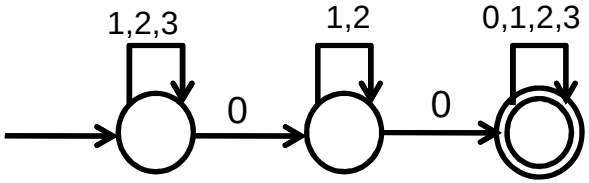
$$L_1 = \{1,2\}^* 0$$

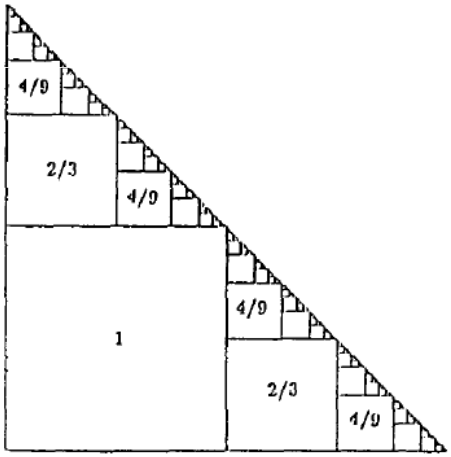
$$L_2 = L_1.\Sigma^* = \{1,2\}^* 0\Sigma^*$$

$$L_3 = \{1,2,3\}^* 0 L_2 = \{1,2,3\}^* 0.\{1,2\}^* 0\Sigma^*$$

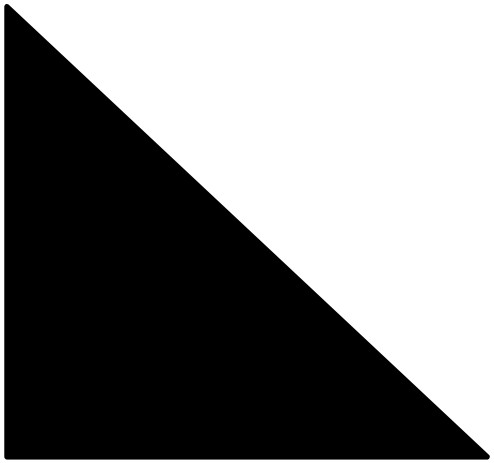


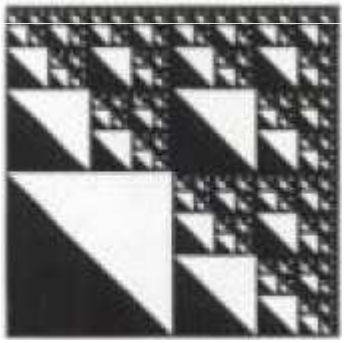




Obdélníky definované daným jazykem





# Konečné automaty - shrnutí

- Mohou generovat jen černobílý obraz
- Generovaný obraz může mít nekonečné rozlišení
- Generovaný obraz může být tzv. multiresolution
- Pomocí konečného automatu lze popsat přesně jen obrazy s regulárním a nebo fraktálním charakterem
- Všechny černobílé obrazy mohou být aproximovány pomocí konečného automatu s nějakou chybou
- Existuje algoritmus pro sestavení automatu na základě obrázku [1]

# Obraz ve stupních šedi

- Hodnoty pixelů jsou reálná čísla
- Většinou jsou hodnoty diskretizovány na čísla od 0 do $2^k$-1 (většinou je k = 8)
- Adresování pixelů je stejné jako u černobílého obrazu
- Obraz o konečném rozlišení $2^m$ x $2^m$ vyjadřuje funkce $f : \Sigma^m \rightarrow \Re$
- Multiresolution obraz je vyjádřen funkcí $g : \Sigma^* \rightarrow \Re$
- U obrazu obvykle požadujeme, aby byly úrovně šedi podobné u všech rozlišení,

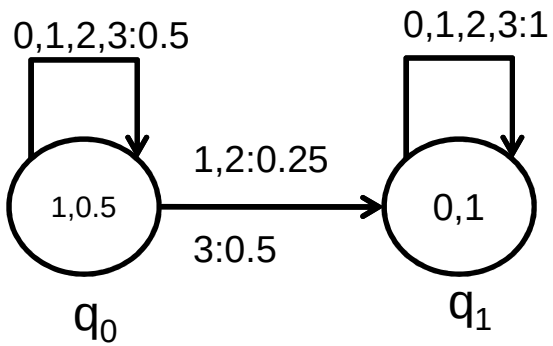
to je zajištěno, pokud je obrazová funkce average preserving

# WFA - příklad

Mějme WFA A:

$$A = \left( \{q_0, q_1\}, \{0,1,2,3\}, (1,0), \left(\frac{1}{2}, 1\right), \begin{Bmatrix} w(q_0,0,q_0)=0.5, w(q_0,1,q_0)=0.5, w(q_0,2,q_0)=0.5, w(q_0,3,q_0)=0.5, \\ w(q_0,1,q_1)=0.25, w(q_0,2,q_1)=0.25, w(q_0,3,q_1)=0.5, \\ w(q_1,0,q_1)=1, w(q_1,1,q_1)=1, w(q_1,2,q_1)=1, w(q_1,3,q_1)=1 \end{Bmatrix} \right)$$

Diagram A:




Výpočet obrázku pro rozlišení *2 x 2* pixelů:

$$f_A(0) = \delta_A(q_0,0) + \delta_A(q_1,0) = 1 \cdot 0.5 \cdot 0.5 + 0 \cdot 1 \cdot 1 = 0.25$$

$$f_A(1) = \delta_A(q_0,1) + \delta_A(q_1,1) = 1 \cdot 0.5 \cdot 0.5 + 1 \cdot 0.25 \cdot 1 + 0 = 0.5$$

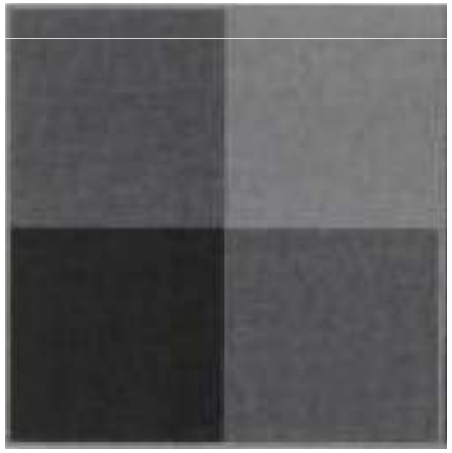
$$f_A(2) = \delta_A(q_0,2) + \delta_A(q_1,2) = 1 \cdot 0.5 \cdot 0.5 + 1 \cdot 0.25 \cdot 1 + 0 = 0.5$$

$$f_A(3) = \delta_A(q_0,3) + \delta_A(q_1,3) = 1 \cdot 0.5 \cdot 0.5 + 1 \cdot 0.5 \cdot 1 + 0 = 1$$

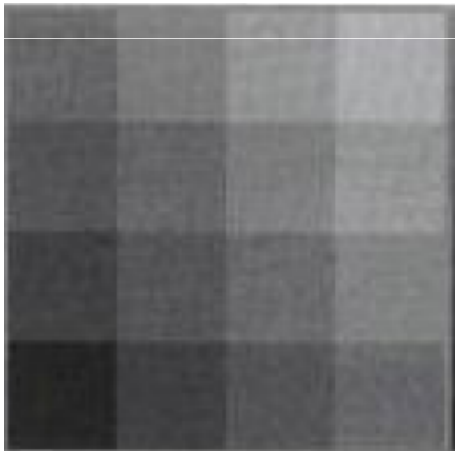
# WFA – příklad pokračování

Multiresolution obrázky generované automatem A:

*2 x 2*



*4 x 4*



*16 x 16*

*256 x 256*

WFA A je average preserving a generuje lineární přechod od černé barvy k bílé

# Operace s WFA

**Dekodování (dekodér):**

- Velmi jednoduché
- Vyčíslení všech řetězců dané délky nad danou abecedou (sestavení obrázku)
- Viz předchozí příklad

**Zoom:**

- Získání vybrané oblasti z celku
- Velmi jednoduchá operace – velmi podobná dekódování
- Má význam u mutliresolution obrázků

- Nechť $f$ je obrázek nad abecedou $\Sigma$, $u \in \Sigma^*$ je slovo a $f_u$ je obrázek získaný pomocí zoomování

$$f_u(w) = f(uw), \text{ pro každé } w \in \Sigma^*$$

- $f_u$ je část obrázku získaná z $f$ zazoomováním na čtverec s adresou $u$
- Jednoduše můžeme sestavit WFA $A_u$, který bude generovat zazoomovanou část obrázku, tak že vezmeme původní WFA A a nahradíme $I$ (počáteční váhy stavů) za $I_u = IW_{a_1}W_{a_2} \ldots W_{a_n}$, kde $u=a_1a_k\ldots a_k$

*$W_{a_1}$ je matice vah přechodů pro symbol $a_1$.*

# Operace s WFA

**Kódování (kodér):**

- Pro jakýkoliv obrázek je možné s jistou chybou sestavit WFA.
- Čím menší chybu požadujeme, tím větší bude výsledný WFA

**Barevné obrázky:**

- Barevný obrázek kódujeme pomocí YIQ - intenzita, odstín, saturace
- Pro každou složku sestavíme jeden WFA

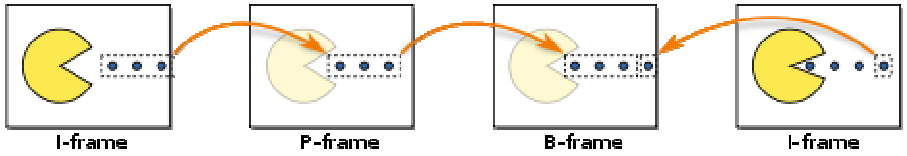
Originální obrázek

Rekonstruovaný obrázek

# WFA - aplikace

**Kódování videa:**

- Video sekvence se rozděluje do pravidelných skupin snímků (GOP-Group of picture)
  - I-frame – vnitřně kódovaný snímek (ukládá se celý snímek)
  - P-frame – predikovaný snímek (ukládá se jen diference)
  - B-frame – dvojitě-predikovaný snímek (ukládá se diference mezi P-Frame a I-frame)

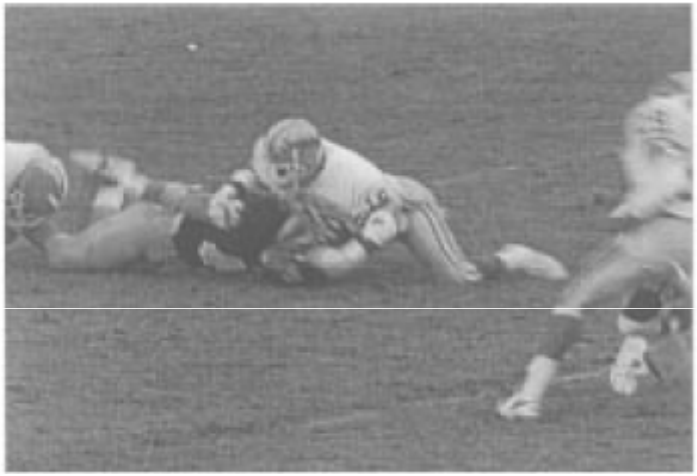


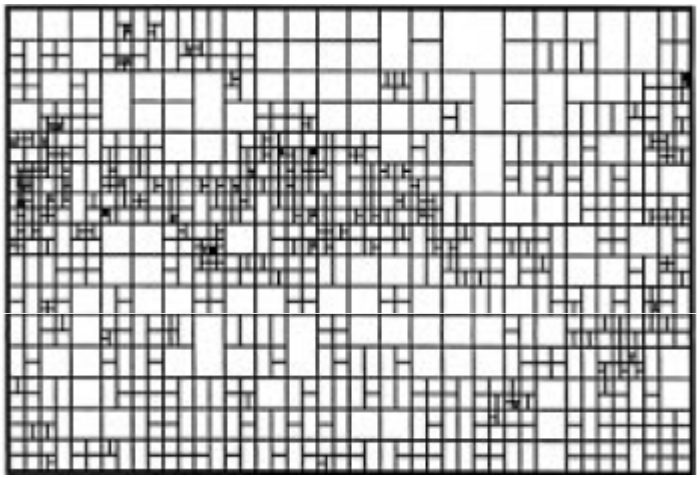

- V plném rozlišení se ukládá jen I-frame, ostatní jsou již komprimovány z podstaty formátu
- Využití komprese založené na WFA pro I-frame
  - Standardně se kóduje v JPEG, použitím WFA se komprimují I-frame 2 krát úsporněji než použitím JPEG komprese

# WFA - aplikace

**Kódování videa:**

- WFA používá adaptivní dělení obrázku do nepřekrývajících se bloků a je sestaven odpovídající WFA automat, který daný obrázek aproximuje

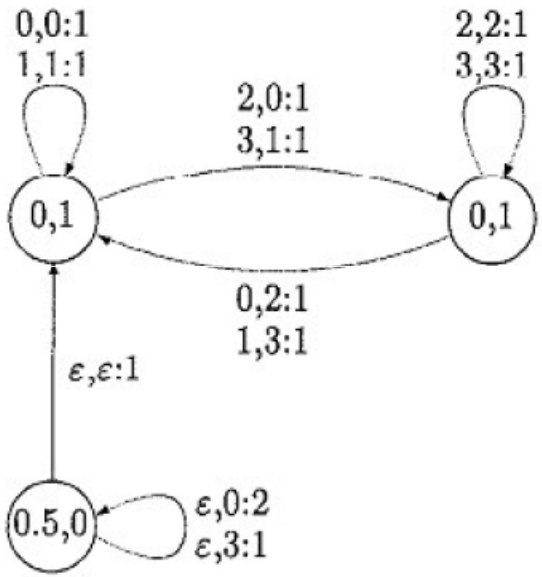




# WFT ve zpracování obrazu

- Téměř všechny transformace obrazu zahrnují změnu intenzity pixelu (barvy)
- Často provádíme afinní transformace, filtrování, výpočet waveletové transformace a jiné
- Takovéto operace mohou být reprezentovány pomocí WFT

WTF provádějící operaci stlačení obrazu a fraktálové kopírování, pro vstup ze stany 23



# WFA & WFT- shrnutí

- Generují obraz většinou ve stupních šedi
- Mohou zpracovávat i barevný obraz
- Generovaný obraz může být multiresolution
- Mohou se provádět operace typu zoom, komprese obrazových dat …
- Operace nad automaty - konkatenace, sloučení …
- Lze popsat libovolný obrázek (obvykle s nějakou chybou)
- Existuje algoritmus pro sestavení automatu na základě obrázku [1]

# Literatura


[1] Rozenberg, G., Salomaa, A.: HandBook of formal languages, vol. 3, ISBN 3-540-60649-1, 1997.

[2] Culik, K., Kari, J.: Digital Images and Formal Languages, HandBook of formal languages, p. 600-616.

[3] Gruska, J.: Foundations of Computing, ISBN 1-85032-243-0, 1997.

[4] Culik, K., Friš. I.: Weighted finite transducers in image processing, Discrete Applied Mathematics 58, 223-237.

[5] Culik, K., Kari, J.: Finite state transformations of images, Proceeding of ICALP 95, Lexture Notes in Computer Science, 944 51-62, Springer-Verlag, Berlin.

[6] Culik, K., Kari, J.: Image Compression Using Weighted Finite Automata, Computer and Graphics 17, 305-313.

[7] Hafner, U., Albert, J., Frank, S: Weighted Finite Automata for Video Compression, IEEE Jornal on selected areas in communications, vol. 16, no. 1, 1998

# Děkuji za pozornost

**Otázky?**