Hopcroft's automaton minimization algorithm and Sturmian words

Jean Berstel, Luc Boasson, Olivier Carton

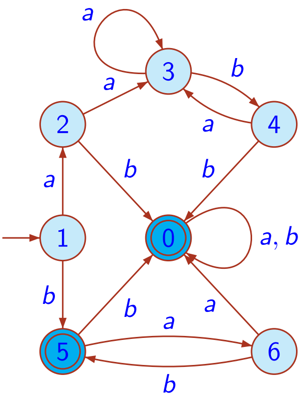
Institut Gaspard-Monge, Université Paris-Est
Liafa, Université Paris Diderot

DMTCS'08

Outline

1. Minimal automaton
   - Minimal automata
2. Hopcroft's algorithm
   - The algorithm
   - Cyclic automata
3. Standard words
   - Definition
   - Standard words and Hopcroft's algorithm
4. Generating series
   - The equation
   - Example: Fibonacci
   - Acceleration
5. Closed form
   - Circular continuant polynomials
6. Further results

Automata



Each state q defines a language $L_q = \{w \mid q \cdot w \text{ is final}\}$.

The automaton is minimal if all languages L_q are distinct.

Here $L_2 = L_4$. States 2 and 4 are (Nerode) equivalent.

The Nerode equivalence gives the coarsest partition that is compatible with the next-state function.

Refinement algorithm

Starts with the partition into two classes 05 and 12346.

A first refinement: $12346 \rightarrow 1234|6$ because of a.

A second refinement: $05 \rightarrow 0|5$ because of a.

History

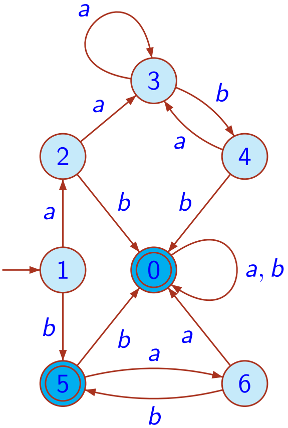
- Hopcroft has developed in 1970 a minimization algorithm that runs in time $O(n \log n)$ on an n state automaton (discarding the alphabet).
- No faster algorithm is known for general automata.
- Question: is the time estimation sharp ?
- A first answer, by Berstel and Carton: there exist automata where you need $\Omega(n \log n)$ steps if you are "unlucky". These are related to De Bruijn words.
- A better answer, by Castiglione, Restivo and Sciortino: there exist automata where you need always $\Omega(n \log n)$ steps. These are related to Fibonacci words.
- Here: the same holds for all Sturmian words corresponding to quadratic irrational slopes.
- Later: Hopcroft's algorithm needs always $\Omega(n \log n)$ steps for all Sturmian words with bounded directive sequence, and it may require less steps.

$\mathcal{P} \leftarrow \{F, F^c\}$ ▷ Initialize current partition $\mathcal{P}$
for all $a \in A$ **do**
  $\text{ADD}((\min(F, F^c), a), \mathcal{W})$ ▷ Initialize waiting set $\mathcal{W}$
while $\mathcal{W} \neq \emptyset$ **do**
  $(C, a) \leftarrow \text{SOME}(\mathcal{W})$ ▷ takes some element in $\mathcal{W}$
  for each $B \in \mathcal{P}$ split by (C, a) **do**
    $B', B'' \leftarrow \text{SPLIT}(B, C, a)$
    REPLACE B by B' and B'' in $\mathcal{P}$
    for all $b \in A$ **do**
      if $(B, b) \in \mathcal{W}$ **then**
        REPLACE (B, b) by (B', b) and (B'', b) in $\mathcal{W}$
      else
        $\text{ADD}((\min(B', B''), b), \mathcal{W})$

Definition

The pair (C, a) splits the set B if both sets $(B \cdot a) \cap C$ and $(B \cdot a) \cap C^c$ are nonempty.

Example

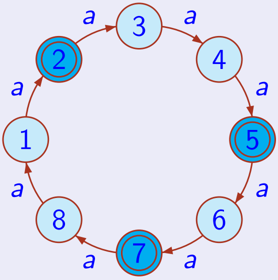


Initiale partition $\mathcal{P}$:	$05 \mid 12346$
Waiting set $\mathcal{W}$:	$(05, a), (05, b)$
Pair chosen :	$(05, a)$
States in inverse :	06
Class to split:	$12346 \to 1234 \mid 6$
Pairs to add :	$(6, a)$ and $(6, b)$
Class to split :	$05 \to 0 \mid 5$
Pair to add:	$(5, a)$ (or $(0, a)$)
Pair to replace:	$(05, b)$: by $(0, b)$ and $(5, b)$
New partition $\mathcal{P}$:	$0 \mid 1234 \mid 5 \mid 6$
New waiting set $\mathcal{W}$:	$(0, b), (6, a), (6, b), (5, a), (5, b)$

Basic fact

Splitting all sets of the current partition by one block (C, a) has a total cost of $\mathrm{Card}(a^{-1}C)$.

Cyclic automaton $\mathcal{A}_w$ for $w = 01001010$



States: $Q = \{1, 2, \ldots, |w|\}$

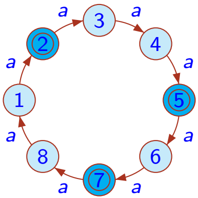
One letter alphabet: $A = \{a\}$

Transitions: $\{k \xrightarrow{a} k+1 \mid k < |w|\} \cup \{|w| \xrightarrow{a} 1\}$

Final states: $F = \{k \mid w_k = 1\}$

Notation

Q_u is the set of starting positions of the occurrences of u in w.



Initiale partition $\mathcal{P}$:	$Q_0 = 13468$, $Q_1 = 257$
Waiting set $\mathcal{W}$:	Q_1
States in inverse of Q_1:	146
Class to split:	$Q_0 = 13468 \to Q_{01} = 146$, $Q_{00} = 38$
New waiting set $\mathcal{W}$:	Q_{00}
New partition $\mathcal{P}$:	$Q_{00} = 38$, $Q_{01} = 146$, $Q_1 = Q_{10} = 257$
States in inverse of Q_{00}:	27
Class to split:	$Q_{10} = 257 \to Q_{100} = 27$, $Q_{101} = 5$
New waiting set $\mathcal{W}$:	Q_{100}
New partition $\mathcal{P}$:	$Q_{001} = 38$, $Q_{010} = 146$, $Q_{100} = 27$, $Q_{101} = 5$

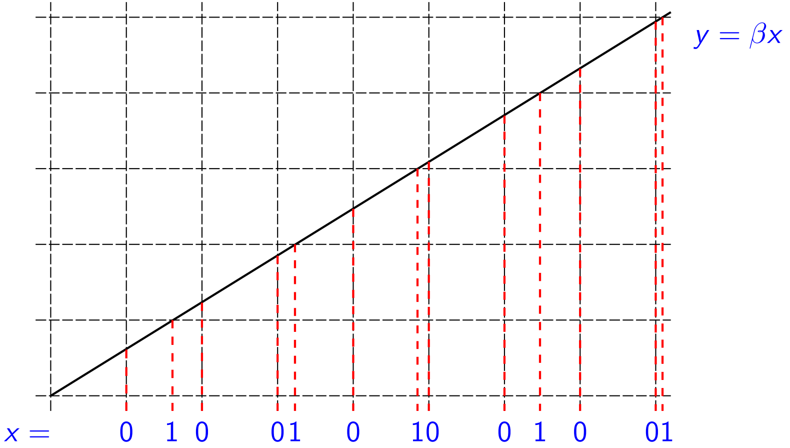
Definition and examples

- directive sequence $d = (d_1, d_2, d_3, \ldots)$ sequence of positive integers
- standard words s_n of binary words defined by $s_0 = 1, s_1 = 0$ and

$$s_{n+1} = s_n^{d_n} s_{n-1} \quad (n \geq 1).$$

- For $d = (\overline{1})$, one gets the Fibonacci words: $s_0 = 1, s_1 = 0, s_2 = 01, s_3 = 010$, $s_4 = 01001$, $s_5 = 01001010$, $s_6 = 0100101001001$, ...
- For $d = (\overline{2, 3})$, one gets $s_0 = 1, s_1 = 0, s_2 = 001, s_3 = 0010010010$,...

Characterization: cutting sequences



Proposition

The standard words converge to the cutting sequence of a straight line $y = \beta x$ *with the irrational slope* $\beta = [0, d_1, d_2, d_3, \ldots]$.

Standard words and Hopcroft's algorithm

Theorem (Castiglione, Restivo, Sciortino)

Let w be a standard word.

- *Hopcroft's algorithm on the cyclic automaton $\mathcal{A}_w$ is uniquely determined.*
- *At each step i of the execution, the current partition is composed if the $i+1$ classes Q_u indexed by the circular factors of length i, and the waiting set is a singleton.*
- *This singleton is the smaller of the sets Q_{u0}, Q_{u1}, where u is the unique circular special factor of length $i-1$.*

Corollary

Let $(s_n)_{n\geq 0}$ be a standard sequence. Then the complexity of Hopcroft's algorithm on the automaton $\mathcal{A}_{s_n}$ is proportional to $\|s_n\|$, where $\|w\| = \sum_{u\in CF(w)} \min(|w|_{u0}, |w|_{u1})$.

Main result

Theorem

Let $(s_n)_{n\geq 0}$ *be the standard sequence defined by an ultimately periodic directive sequence* d*. Then* $\|s_n\| = \Theta(n|s_n|)$*, and the complexity of Hopcroft's algorithm on the automata* $\mathcal{A}_{s_n}$ *is in* $\Theta(N \log N)$ *with* $N = |s_n|$*.*

Let $d = (d_1, d_2, \dots)$ and $(s_n)_{n\geq 0}$ be the standard sequence defined by d.
Set $a_n = |s_n|_1$ and $c_n = \|s_n\| = \sum_{u\in CF(s_n)} \min(|s_n|_{u0}, |s_n|_{u1})$.

c_n is the complexity of Hopcroft's algorithm on $\mathcal{A}_{s_n}$, and a_n is the size of $\mathcal{A}_{s_n}$.

The generating series are $A_d(x) = \sum_{n\geq 1} a_n x^n$, $\quad C_d(x) = \sum_{n\geq 0} c_n x^n$.

Proposition

For any directive sequence $d = (d_1, d_2, \dots)$, one has

$$C_d(x) = A_d(x) + x^{\delta(d)} C_{\tau(d)}(x) + x^{1+\delta(T(d))} C_{\tau(T(d))}(x)\,.$$

$$\tau(d) = \begin{cases} (d_1 - 1, d_2, d_3, \dots) & \text{if } d_1 > 1 \\ (d_2, d_3, \dots) & \text{otherwise}. \end{cases} \qquad \delta(d) = \begin{cases} 0 & \text{if } d_1 > 1\,, \\ 1 & \text{otherwise.} \end{cases}$$

and $T(d) = \tau^{d_1}(d) = (d_2, d_3, \dots)$.

Example: Fibonacci

For $d = (\bar{1})$, one has $\tau(d) = T(d) = d$, and $\delta(d) = 1$. The equation becomes

$$C_d(x) = A_d(x) + (x + x^2)C_d(x)\,,$$

from which we get $C_d(x) = \dfrac{A_d(x)}{1 - x - x^2}$. Clearly $a_{n+2} = a_{n+1} + a_n$ for $n \geq 0$, and since $a_0 = 1$ and $a_1 = 0$, one gets $A_d(x) = \dfrac{x^2}{1 - x - x^2}$. Thus

$$C_d(x) = \frac{x^2}{(1 - x - x^2)^2}\,.$$

This proves that $c_n \sim Cn\varphi^n$, where φ is the golden ratio, and proves the theorem of Castiglione, Restivo and Sciortino.

Another example $d = (\overline{2,3})$

$$\begin{aligned}
C_{(\overline{2,3})} &= A_{(\overline{2,3})} + C_{(1,\overline{3,2})} + xC_{(2,\overline{2,3})} \\
C_{(1,\overline{3,2})} &= A_{(1,\overline{3,2})} + xC_{(\overline{3,2})} + xC_{(2,\overline{2,3})} \\
C_{(2,\overline{2,3})} &= A_{(2,\overline{2,3})} + C_{(1,\overline{2,3})} + xC_{(1,\overline{3,2})} \\
C_{(\overline{3,2})} &= A_{(\overline{3,2})} + C_{(2,\overline{2,3})} + xC_{(1,\overline{3,2})} \\
C_{(1,\overline{2,3})} &= A_{(1,\overline{2,3})} + xC_{(\overline{2,3})} + xC_{(1,\overline{3,2})}
\end{aligned}$$

Here $A_{(\overline{2,3})} = A_{(1,\overline{3,2})}$ and $A_{(\overline{3,2})} = A_{(2,\overline{2,3})} = A_{(1,\overline{2,3})}$.
Set $D_1 = C_{(1,\overline{3,2})}$ and $D_2 = C_{(2,\overline{2,3})}$.

$$C_{(\overline{2,3})} = A_{(\overline{2,3})} + D_1 + xD_2\,,$$

where D_1 and D_2 satisfy the equations

$$\begin{aligned}
D_1 &= A_{(\overline{2,3})} + xA_{(\overline{3,2})} + 2xD_2 + x^2D_1 \\
D_2 &= 2A_{(\overline{3,2})} + xA_{(\overline{2,3})} + 3xD_1 + x^2D_2\,.
\end{aligned}$$

Thus the original system of 5 equations in the C_u is replaced by a system of 2 equations in D_1 and D_2.

Let $d = (d_1, d_2, \ldots)$ be a directive sequence, and for $i \geq 1$, set

$$e_i = T^{i-1}(d) = (d_i, d_{i+1}, \ldots)\,.$$

Set also

$$D_i = x^{\delta(e_i)} C_{\tau(e_i)}\,, \qquad B_i = (d_i - 1)A_{e_i} + xA_{e_{i+1}}\,.$$

With these notations, the following system of equation holds.

Proposition

The following equations hold

$$\begin{aligned} C_d &= A_d + D_1 + xD_2 \\ D_i &= B_i + d_i x D_{i+1} + x^2 D_{i+2} \qquad (i \geq 1) \end{aligned}$$

Theorem

If d *is a purely periodic directive sequence with period* k, *then*

$$A_d(x) = \sum a_n x^n = x\frac{R(x)}{Q(x)},$$

where $R(x)$ *is a polynomial of degree* $< 2k$ *and*

$$Q(x) = 1 - Z(d_1, \ldots, d_k)x^k + (-1)^k x^{2k}$$

where $Z(x_1, \ldots, x_k)$ *is a polynomial in the variables* $x_1, \ldots, x_k$. *Moreover,* $a_n = \Theta(\rho^n)$, *where* ρ *is the unique real root greater than* 1 *of the reciprocal polynomial of* $Q(x)$. *Next,*

$$C_d(x) = \sum c_n x^n = \frac{S(x)}{Q(x)^2},$$

where $S(x)$ *is a polynomial, and* $c_n = \Theta(n\rho^n)$.

Circular continuant polynomials

Replace in the word $x_1 \cdots x_n$ a factor $x_i x_{i+1}$ of variables with consecutive indices by 1. The replacement of $x_n x_1$ is allowed for circular continuants. The following are the first circular continuant polynomials.

$$\begin{aligned} Z(x_1) &= x_1 \\ Z(x_1, x_2) &= x_1 x_2 + 2 \\ Z(x_1, x_2, x_3) &= x_1 x_2 x_3 + x_1 + x_2 + x_3 \\ Z(x_1, x_2, x_3, x_4) &= x_1 x_2 x_3 x_4 + x_1 x_2 + x_2 x_3 + x_3 x_4 + x_4 x_1 + 2\,. \end{aligned}$$

The first continuant polynomials are

$$\begin{aligned} K(x_1) &= x_1 \\ K(x_1, x_2) &= x_1 x_2 + 1 \\ K(x_1, x_2, x_3) &= x_1 x_2 x_3 + x_1 + x_3 \\ K(x_1, x_2, x_3, x_4) &= x_1 x_2 x_3 x_4 + x_1 x_2 + x_3 x_4 + x_1 x_4 + 1\,. \end{aligned}$$

They are related by

$$Z(x_1, x_2, \ldots, x_n) = K(x_1, x_2, \ldots, x_n) + K(x_2, \ldots, x_{n-1})\,.$$

Further results

Theorem

For any sequence d*, one has* $c_n = \Theta(na_n)$.

Corollary

If a_n *grows at most exponentially, then* $c_n = \Theta(a_n \log a_n)$ *and* $n = \Theta(\log a_n)$.

Corollary

If the elements of the sequence d *are bounded, then* $c_n = \Theta(a_n \log a_n)$.

Corollary

There exist directive sequences d *such that* $c_n = O(a_n \log \log a_n)$.

A combinatorial lemma (one of four)

Lemma

Assume $d_2 > 1$*, and let* t_n *be the sequence of standard words generated by* $\tau T(d) = (d_2 - 1, d_3, d_4, \ldots)$*. Let* β *be the morphism defined by*

$$\beta(0) = 10^{d_1} \text{ and } \beta(1) = 10^{d_1+1}$$

- *Then* $s_{n+1}0^{d_1} = 0^{d_1}\beta(t_n)$ *for* $n \geq 1$*.*
- *If* v *is a circular special factor of* t_n*, then* $\beta(v)10^{d_1}$ *is a circular special factor of* s_{n+1}*.*
- *Conversely, if* w *is a circular special factor of* s_{n+1} *starting with* 1*, then* w *has the form* $w = \beta(v)10^{d_1}$ *for some circular special factor* v *of* t_n*.*
- *Moreover,* $|s_{n+1}|_{w0} = |t_n|_{v1}$ *and* $|s_{n+1}|_{w1} = |t_n|_{v0}$*.*

Example ($d = (\overline{2,3})$, so $\beta(0) = 100$, $\beta(1) = 1000$)

$$
\begin{array}{ll}
t_0 = 1 & s_0 = 1 \\
t_1 = 0 & s_1 = 0 \\
t_2 = 001 & s_2 = 001 \\
t_3 = (001)^2 0 & s_3 = (001)^3
\end{array}
$$

$$s_3 00 = 00.100.100.1000 = 00\beta(001) = 00\beta(t_2)$$

$$t_2 = \underline{001},\ s_3 00 = 00\underline{1001001}000 = 00100\underline{1001000}$$