Continuant polynomials, circular Sturmian words and the worst-case behavior of Hopcroft's automaton minimization algorithm

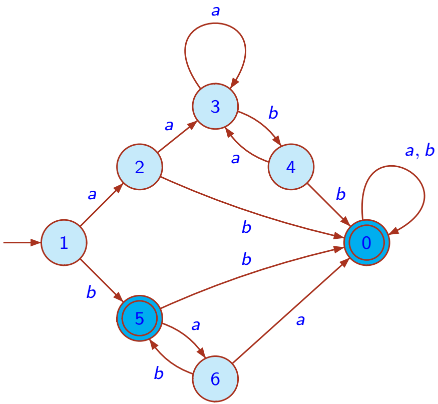
Jean Berstel, Luc Boasson, Olivier Carton

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

Jorcad'08, Rouen 17–19 septembre 2008

Outline

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.

Outline

- 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 (CIAA 2004): 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 (WORDS 2007, LATA 2008): there exist automata where you need always $\Omega(n \log n)$ steps. These are related to Fibonacci words.
- Castiglione, Restivo and Sciortino (TCS) describe deep connections between statistics related to Hopcroft's algorithm and structure of standard words.
- Here: Hopcroft's algorithm needs always $\Omega(n \log n)$ steps for all Sturmian words with bounded directive sequence, and it may require less steps.

The algorithm

HopcroftMinimization()

$\mathcal{P} \leftarrow \{F, F^c\}$
$C \leftarrow \min(F, F^c)$
for $a \in A$ **do**
    Add$((C, a), \mathcal{W})$ ▷ adds (C, a) to set $\mathcal{W}$
while $\mathcal{W} \neq \emptyset$ **do**
    $(C, a) \leftarrow$ Some$(\mathcal{W})$ ▷ takes some element in $\mathcal{W}$
    for each $B \in \mathcal{P}$ split by (C, a) **do**
        $B', B'' \leftarrow$ Split(B, C, a)
        Replace B by B' and B'' in $\mathcal{P}$
        $C \leftarrow \min(B', B'')$
        for $b \in A$ **do**
            if $(B, b) \in \mathcal{W}$ **then**
                Replace (B, b) by (B', b) and (B'', b) in $\mathcal{W}$
            else Add$((C, b), \mathcal{W})$

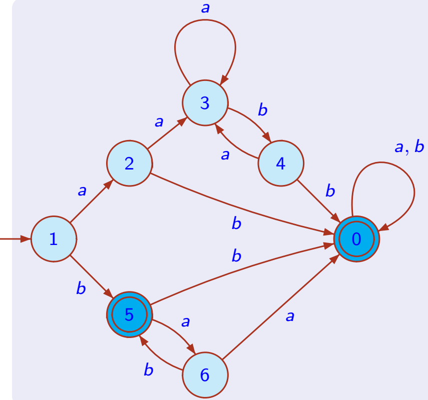
Definition

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.

Notation

$\mathcal{P}$ is the current partition. $\mathcal{W}$ is the waiting set.

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:	$12346 \longrightarrow 1234 \mid 6$
Pairs to add :	$(6, a)$ and $(6, b)$
Class to split :	$05 \longrightarrow 0 \mid 5$
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

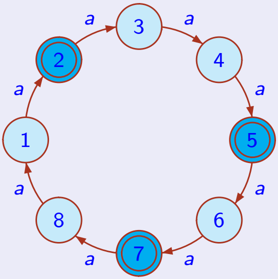
Splitting all sets of the current partition by one block (C, a) has a total cost of $\mathrm{Card}(a^{-1}C)$.

Cyclic automata

Definition

One-letter automaton with states on a unique cycle. The sequence of nonterminal and of terminal states form a circular binary word.

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



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

Notation

Standard words

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.
- For $d = (\overline{2, 3})$, one gets $s_0 = 1, s_1 = 0, s_2 = 001, s_3 = 0010010010, \ldots$

Proposition

A standard word is primitive. If $u01$ is a standard word, then u is a palindrome, $u10$ is standard and $u01$ and $u10$ are conjugate words.

Proposition (Borel, Reutenauer)

A word w is standard if and only if it has exactly $i+1$ circular factors of length i, and exactly one circular special factor for each $i = 0, \ldots, |w| - 2$.

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

$$\|w\| = \sum_{u\in CF(w)} \min(|w|_{u0}, |w|_{u1})\,.$$

Standard words and Hopcroft's algorithm

Example

We compute $\|w\| = \sum_{u \in CF(w)} \min(|w|_{u0}, |w|_{u1})$ for $w = 01001010$.

u	$\|w\|_{u0}$	$\|w\|_{u1}$	min
ε	5	3	3
0	2	3	2
10	2	1	1
010	2	1	1
0010	1	1	1
10010	1	1	1
010010	1	1	1

So the number $\|w\|$ equals 10.

Theorem (Our main result)

Let $(s_n)_{n\geq 0}$ be the standard sequence defined by a directive sequence d with bounded elements. 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 $N = |s_n|$.

Generating series

Let $d = (d_1, d_2, \ldots)$ 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 for s_n, and a_n is (almost) the length of 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 $d = (d_1, d_2, \ldots)$, *one has*

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

Here

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

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

Example: For $d = (1, 2, 3, 4, \ldots)$, one gets $\tau(d) = (2, 3, 4, \ldots)$ and $\delta(d) = 1$.

Example: Fibonacci

For $d = (\overline{1})$, 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. This was proved by Castiglione, Restivo and Sciortino (WORDS'07).

Another example

Proposition

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

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.

Acceleration

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

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

Continuant Polynomials

Definition

The continuant polynomials $K_n(x_1, \ldots, x_n)$, for $n \geq -1$ are a family of polynomials in the variables $x_1, \ldots, x_n$ defined by $K_{-1} = 0$, $K_0 = 1$ and, for $n \geq 1$, by

$$K_n(x_1, \ldots, x_n) = x_1 K_{n-1}(x_2, \ldots, x_n) + K_{n-2}(x_3, \ldots, x_n)\,.$$

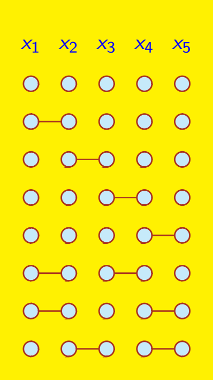
The first continuant polynomials are

$$\begin{aligned}
K_1(x_1) &= x_1\\
K_2(x_1, x_2) &= x_1x_2 + 1\\
K_3(x_1, x_2, x_3) &= x_1x_2x_3 + x_1 + x_3\\
K_4(x_1, x_2, x_3, x_4) &= x_1x_2x_3x_4 + x_1x_2 + x_3x_4 + x_1x_4 + 1\,.
\end{aligned}$$

Combinatorial Interpretation

The Morse code

$$K_5(x_1, x_2, x_3, x_4, x_5) = x_1x_2x_3x_4x_5 + x_3x_4x_5 + x_1x_4x_5 + x_1x_2x_5 + x_1x_2x_3 + x_5 + x_3 + x_1$$



$x_1x_2x_3x_4x_5$

$x_3x_4x_5$

$x_1x_4x_5$

$x_1x_2x_5$

$x_1x_2x_3$

x_5

x_3

x_1

Equivalent definitions

$$K_n(x_1, \ldots, x_n) = x_1 K_{n-1}(x_2, \ldots, x_n) + K_{n-2}(x_3, \ldots, x_n)\,,$$
$$K_n(x_1, \ldots, x_n) = K_{n-1}(x_1, \ldots, x_{n-1})x_n + K_{n-2}(x_1, \ldots, x_{n-2})$$

See Graham, Knuth, Patashnik, *Concrete Mathematics*, for other properties.

Continuant polynomials and continued fractions

Let $d = (d_1, d_2, d_3, \ldots)$ be a sequence of positive numbers. The continued fraction defined by d is denoted $\alpha = [d_1, d_2, d_3, \ldots]$ and is defined by

$$\alpha = d_1 + \cfrac{1}{d_2 + \cfrac{1}{d_3 + \cdots}} \; .$$

The finite initial parts $[d_1, d_2 \ldots, d_n]$ of d define rational numbers

$$d_1 + \cfrac{1}{d_2 + \cfrac{1}{d_3 + \ddots + \cfrac{1}{d_n}}} = \frac{K_n(d_1, \ldots, d_n)}{K_{n-1}(d_2, \ldots, d_n)} \; .$$

Continuant polynomials and standard words

One has

$$a_{n+2} = K_n(d_2, \ldots, d_{n+1}) \quad (n \geq -1)$$

and

$$A_d(x) = x^2 \sum_{n \geq 0} K_n(d_2, \ldots, d_{n+1}) x^n .$$

The series C_d also has an expression with continuants

$$C_d = x^2 \sum_{n \geq 0} (K_n(d_2, \ldots, d_{n+1}) + N_{n+1}(d_1, \ldots, d_{n+1}) + N_n(d_1, \ldots, d_n)) x^n .$$

where

$$L_n(x_1, \ldots, x_n) = K_n(x_1, \ldots, x_n) - K_{n-1}(x_2, \ldots, x_n) .$$

$$N_n(x_1, \ldots, x_n) = \sum_{i=0}^{n-1} K_i(x_1, \ldots, x_i) L_{n-i}(x_{i+1}, \ldots, x_n) .$$

Theorem

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

It suffices to show that $N_n(d_1, \ldots, d_n) = \Theta(nK_n(d_1, \ldots, d_n))$.

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 $c_n = O(a_n \log \log a_n)$.