VC-dimensions of nondeterministic finite automata for words of equal length*

Bjørn Kjos-Hanssen[1]
Clyde James Felix, Sun Young Kim,
Ethan Lamb, Davin Takahashi
[1]Department of Mathematics
University of Hawai‘i at Mānoa, 2565 McCarthy Mall
Honolulu, Hawai‘i 96822
bjoern.kjos-hanssen@hawaii.edu

Abstract

Ishigami and Tani studied VC-dimensions of deterministic finite automata. We obtain analogous results for the nondeterministic case by extending a result of Champarnaud and Pin, who proved that the maximal deterministic state complexity of a set of binary words of length n is

$$\sum_{i=0}^{n} \min(2^i, 2^{2^{n-i}} - 1).$$

We show that for the nondeterministic case, if we fully restrict attention to words of length n, then we at most need the strictly increasing initial terms in this sum.

Introduction

Consider the set of points $F = \{(-1,0),(0,0),(1,1)\}$. We can separate $\{(1,1\}$ from $\{(-1,0),(0,0)\}$ by the straight line $y = x/2$. In general, such a separation can be made provided the three points do not all lie on the same line. However, with four points a new difficulty arises. Consider the set of points

$$\{(0,0),(0,1),(1,0),(1,1)\}.$$

There is no straight line that separates $\{(1,0),(0,1)\}$ from $\{(0,0),(1,1)\}$. There is no way to get $(1,0)$ and $(0,1)$ to lie on one side of the separating line, and $(0,0)$ and $(1,1)$ on the other. This can be summarized by saying that our statistical model, involving finding parameters (w_1, w_2, b) such that $w_1 x + w_2 y = b$, has a *Vapnik–Chervonenkis dimension* that is at least 3, but not at least 4. In other words, the VC dimension is exactly 3. A set of three points can be *shattered* but some sets of four points cannot.

For this model, no set of size 4 can be shattered. This may be demonstrated by consider two cases.

1. No three points lie on the same line and the four points form the corners of a convex quadrilateral. In this case, the two points diagonally across from each other cannot be separated from the other two.
2. One point z lies inside the convex hull of the other points w, x, y. In this case, x cannot be separated from $\{w, x, y\}$.

The observation that existence of a shattered set may often lead to *every* set being shattered, leads us to define a notion of *lower VC-dimension*.

This notion will in particular be relevant in the setting of finite automata, where we will obtain our main results.

As usual we let $L(M)$ denote the language accepted by the automaton M.

Definition 1. We define the (upper) (n,q)-VC dimension of NFA (nondeterministic finite automata), $VC^{\text{upper}}_{n,q}$, to be the largest number m such that there are distinct binary words $x_1, \dots, x_m$ of length n such that for each set $F \subseteq \{1, \dots, m\}$, there is an NFA M with q states, such that

$$L(M) \cap \{x_1, \dots, x_m\} = F.$$

We say that $\{x_1, \dots, x_m\}$ is *shattered*. The lower (n,q)-VC dimension of NFA, $VC^{\text{lower}}_{n,q}$, is defined similarly, but replacing an existential quantifier by a universal one: it is the largest number m such that for all distinct binary words $x_1, \dots, x_m$ of length n, and for each set $F \subseteq \{1, \dots, m\}$, there is an NFA M such that $L(M) \cap \{x_1, \dots, x_m\} = F$.

More generally, the lower VC-dimension of a set of automata $\mathcal{S}$ for a set of words $\mathcal{W}$ is the largest number m such that for all distinct words $x_1, \dots, x_m$ and $F \subseteq \{x_1, \dots, x_m\} \subseteq \mathcal{W}$, there is an $M \in \mathcal{S}$ such that $L(M) \cap \{x_1, \dots, x_m\} = F$.

Theorem 2. *There exists a set of automata $\mathcal{S}$ such that the lower VC-dimension of $\mathcal{S}$ for $\{0,1\}^*$ is finite and the upper VC-dimension of $\mathcal{S}$ for $\{0,1\}^*$ is infinite.*

Proof. Let $\mathcal{S}$ consist of NFAs with no restriction on the number of states, but using a unary alphabet $\Sigma = \{0\}$. In this case the upper VC-dimension is infinite, as these automata can shatter $\{0^k : k \leq n\}$ for any n. The lower VC-dimension is 0 as these NFAs cannot accept $\{1\}$. ☐

*The authors were supported by *Faculty Mentoring Grants for Summer Undergraduate Research and Creative Works* , sponsored by the Undergraduate Research Opportunities Program (UROP) in the Office of the Vice Chancellor for Research, University of Hawai‘i. This work was partially supported by a grant from the Simons Foundation (#315188 to Bjørn Kjos-Hanssen).

Computational complexity of upper and lower VC-dimension. In a way an example of non-shatterability is more intelligible since it gives a specific set and a specific subset that cannot be achieved by any automaton, where to witness shatterability we need to give a long list of automata. In this sense, lower VC-dimension is one level lower in the computational complexity hierarchy and more amenable to direct study. On the other hand, the upper VC-dimension is the one that is important for learning theory via the Glivenko–Cantelli theorem.

Remark 1. The (4,2)-VC dimension of NFA is at least 3, because you can take $X = \{x_1, x_2, x_3\} = \{0000, 0101, 1000\}$. Each singleton F is fairly easy to find an M for in this case, and for the sets F of size 2, we can note that $\{0000, 0101\}$ consists of the elements of X starting with "0"; $\{0000, 1000\}$ similarly ending with 0; and $\{0101, 1000\}$ are the elements of X containing at least one "1".

The choice $X = \{0000, 1111, 0101\}$ on the other hand does not seem to work. So we are interested in which choices of X are suitable in general, and how this relates to automatic complexity. The complexity of a set S of words of a given length, studied by Champarnaud and Pin [2], Campeanu and Ho [1], and Kjos-Hanssen and Liu [7], is related to the shatterability of S. In fact, Ishigami and Tani [5] use the result of Champarnaud and Pin to show that the VC-dimension of DFA for 2 letters is $\Theta(n \log n)$. Ishigami and Tani's result applies to lower VC-dimension as well.

Definition 3. The deterministic state complexity of a set of binary words F of length n is the minimum number of states $s(F)$ of a DFA M such that $L(M) = F$.

Theorem 4 (Champarnaud and Pin [2])**.**

$$\max_{F \subseteq \{0,1\}^n} s(F) = \sum_{i=0}^{n} \min(2^i, 2^{2^{n-i}} - 1).$$

Theorem 4 does not require that all words have the same length, but it will be important for our nondeterministic version in Theorem 10. For instance,

$$\begin{aligned} q_0 &= 1, \\ q_1 &= \min(2^0, 2^{2^{1-0}} - 1) + \min(2^1, 2^{2^{1-1}} - 1) = 2, \\ q_2 &= 1 + \min(2^1, 2^{2^{2-1}} - 1) + 1 = 4. \end{aligned}$$

The equation $q_2 = 4$ tells us that with 4 states, we can implement any function on words of length 2 and hence we can shatter $\{00, 01, 10, 11\}$. In other words, the VC-dimension at $q = 4$ and $n = 2$ is at least 4 for (non-total) DFAs. Next,

$$q_3 = 1 + 2 + 3 + 1 = 7.$$

This tells us that the $(n = 3, q = 7)$ VC-dimension is at least $2^3 = 8$. The sequence continues as in Table 1. In each case the (n, q)-VC dimension is at least 2^n for $q \geq \max_{F \subseteq \{0,1\}^n} s(F)$.

The fact that the $n = 5$, $q = 14$ entry in Table 2 is at least 24 follows from Theorem 13. Similarly the $n = 4$, $q = 6$ entry follows from Theorem 12.

n	q		
0	1	1	
1	2	1	+1
2	4	1+2	+1
3	7	1+2	+3+1
4	11	1+2+4	+3+1
5	19	1+2+4+8	+3+1
6	34	1+2+4+8	+15+3+1
7	50	1+2+4+8+16	+15+3+1
8	82	1+2+4+8+16+32	+15+3+1

Table 1: Champarnaud–Pin bounds.

Connection to the Separating Words problem. This problem has been studied by Shallit and others [3] and concerns shattering of a set of size 2. The problem is well described at [10]: we let $\text{sep}(w, x)$ be the smallest number of states in a DFA that accepts one of w, x but not the other. Then we let

$$S(n) = \max_{|w|=|x|=n w \neq x} \text{sep}(w, x).$$

Computing $S(n)$ is then equivalently described as follows: Given n, for which q is the lower VC-dimension above 2? Robson [9] showed that $S(n) = O(n^{2/5} (\log n)^{3/5})$.

Shattering sets of long words

Inspection of Table 2 suggests the following.

Conjecture 1. The upper VC-dimension of NFA is non-decreasing in the word length n for a fixed number of states q.

Informally, Conjecture 1 says that automata with a fixed number of states can shatter sets of *long* words just as well as they can shatter sets of *short* words. A possible counterexample, subject to calculation, is $n = 5, q = 3$ vs. $n = 4, q = 3$. By adding an extra bit like 0 we can at least conclude that $VC^{\text{upper}}_{n,q} \leq VC^{\text{upper}}_{n+1,q+1}$.

Definition 5. For a nonnegative integer k, we let $[k] = \{1, \ldots, k\}$. For a fixed q let us say that two sequences of strings $x_1, \ldots, x_k$ and $y_1, \ldots, y_k$ are $\approx_q$-similar if for each $F \subseteq [k]$ there is an automaton M with q states that accepts all the words x_i, $i \in F$ and y_i, $i \in F$, but none of the words x_i, $i \notin F$ and y_i, $i \notin F$. (We may allow changing the accept states here.) Let us say that two sets of strings are $\sim$-similar if there are orderings of them that are $\approx$-similar. That is, the sets of strings are shattered by automata with q states for "the same reason".

One way to prove that $VC^{\text{upper}}_{n,q} \leq VC^{\text{upper}}_{n+1,q}$ would be if each shattered set of strings of length n is similar to a shattered sets of strings of length $n + 1$.

For instance, when $q = 1$ we have $\{0^n, 1^n\} \sim \{0^{n+1}, 1^{n+1}\}$ because $(0^n, 1^n) \approx (0^{n+1}, 1^{n+1})$.

And, to show that at $q = 2$, $(00, 01, 10, 11) \not\approx (000, 001, 010, 011)$, we could proceed by trying to show that there is no 2-state NFA that accepts all of $\{00, 01, 10, 000, 001, 010\}$ but none of $\{11, 011\}$.

q										
∞	1/1	2/2	4/4	8/8	16/16	32/32	64/64	...		
...	...	...	...	...						
15	1/1	2/2	4/4	8/8	16/16	32/**32**	?	...		
14	1/1	2/2	4/4	8/8	16/16	(≥ 24)/?	?	...		
...	...	...	...	...						
7	1/1	2/2	4/4	8/8	16/**16**	?	?			
6	1/1	2/2	4/4	8/**8**	≥ 12/?	?	?			
5	1/1	2/2	4/4	8/8	?	?	?			
4	1/1	2/2	4/4	8/8	(≥ 10)/?	?	?			
3	1/1	2/2	4/**4**	8/8	9/5	(≥ 8)/4	(≥ 8)/$[2,4]$	(≥ 7)/$[2,4]$		
2	1/1	2/2	4/4	5/3	5/2	5/2	(≥ 5)/1	(≥ 5)/1		
1	**1**	2/**2**	2/1	2/1	2/1	2/1	2/1	2/1	...	2/1
	0	1	2	3	4	5	6	7	...	∞ n

Table 2: VC-dimension of NFA for various choices of word length n and number of states q, in the form "upper/lower" or just "upper". For $n \geq 3$ and $q \geq 2$ see [8]. **Bold** numbers indicate results derived from Theorem 10.

Combining Hyde and Champarnaud–Pin

Hyde [4] proved that nondeterministic automatic complexity $A_N(x)$ for x of length n is generally at least $\lfloor n/2 \rfloor + 1$. Shallit and Wang originally [11] defined the *automatic complexity* of a finite binary string $x = x_1 \dots x_n$ to be the least number $A_D(x)$ of states of a deterministic finite automaton M such that x is the only string of length n in the language accepted by M.

Definition 6 ([4])**.** The nondeterministic automatic complexity $A_N(w)$ of a word w is the minimum number of states of an NFA M (having no ϵ-transitions) accepting w such that there is only one accepting path in M of length $|w|$.

Theorem 7 (Hyde)**.** *The nondeterministic automatic complexity $A_N(x)$ of a string x of length n satisfies*

$$A_N(x) \leq b(n) := \lfloor n/2 \rfloor + 1.$$

The proof for the case of an odd $n = 2k + 1$ is in outline that there is an automaton with $k + 1$ states $q_0, \dots q_k$, that accepts a word of length n uniquely along the path

$$q_0, q_1, \dots, q_{k-1}, q_k, q_k, q_{k-1}, q_{k-2}, \dots, q_1, q_0.$$

Definition 8. Let a, t be nonnegative integers and let $k = t + a$. The automaton $M_{t,a}$ has states $0, 1, \dots, k$, start state 0, final state 0, and the following edges: $(i, i+1)$ for each $0 \leq i < k$, (k, k), $(i, i-1)$ for each $t + 1 \leq i \leq k$, and $(t, 0)$.

The automaton $M_{t,a}$ is shown in Figure 1.

Lemma 1. Let a, t be nonnegative integers and $n = t + 2a + 2$. Suppose that $t \geq \lfloor n/2 \rfloor$. Then $M_{t,a}$ accepts a unique word of length n, along a unique accepting path.

Proof. Since 0 is both the start and accept state, we must traverse the length $t + 1$ cycle $(0, 1, \dots, t-1, t, 0)$ at least once. If we traverse it twice then

$$n \geq 2(t+1) \geq 2(\lfloor n/2 \rfloor + 1) \geq 2((n/2 - 1/2) + 1) = n + 1,$$

a contradiction. This means that $n - (t+1)$ symbols may be read during the remainder of an accepting run. Let $k = t + a$. Since $n = 2(k+1) - t$, it follows that $n - t$ is even and $n - (t+1)$ is odd. This forces us to accept only along a single path, as in Hyde's Theorem 7. □

Theorem 10 is a nondeterministic version of Theorem 4. The difference is that we include only the strictly increasing initial terms from Theorem 4.

Definition 9. The length-restricted nondeterministic state complexity of a set $F \subseteq \{0,1\}^n$ is the minimum number of states $s'_N(F)$ of an NFA M such that $L(M) \cap \{0,1\}^n = F$.

Theorem 10 (Main theorem)**.** *Let n be a nonnegative integer and $F \subseteq \{0,1\}^n$, then*

$$s'_N(F) \leq \sum_{i=0}^{k} a_i, \quad \textit{where } a_i = \min(2^i, 2^{2^{n-i}} - 1),$$

where k is greatest such that $a_{k-1} < a_k$.

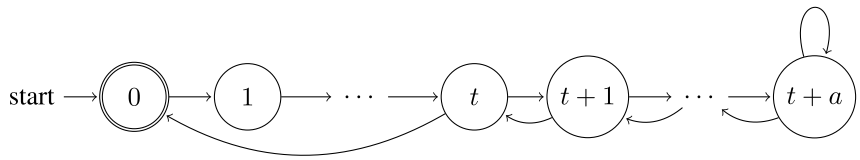


Figure 1: The automaton $M_{t,a}$ from Definition 8.

Proof. For a family of sets $\mathcal{F} \subseteq \{0,1\}^n$, we construct a binary tree and then turn around and go back when $\mathcal{F}$ is "small enough". See Figure 3 for an illustration of the idea.

The rapid decrease of the sequence $2^{2^{n-i}} - 1$ is also used in the proof, to ensure that no "shortcuts" can occur. Consider the case $n = 6$. We arrange the states in columns of 1,2,4,8,15 and then let the 3 penultimate states be chosen from among the column of 15. The general pattern is illustrated by the following example for $n = 21$. In Figure 2 we are showing one state per column (i.e., level of the tree) and connecting column i to column j if some state in column i has an edge to column j. The displayed states are labeled by the number of states of the automaton in the corresponding column.

Here "on the way back to the start state (column 0)" we are using

- at most $2^{2^4} - 1$ of the 2^{16} available states in column 16,
- at most $2^{2^3} - 1$ of the 2^{15} available states in column 15,
- at most $2^{2^2} - 1$ of the 2^{14} available states in column 14, and
- at most $2^{2^1} - 1 = 3$ of the 2^{13} available states in column 13.

We have a cycle of more than half the length followed by a Hyde–style module, so that the columns form an automaton of the form $M_{t,a}$ (Definition 8). This ensures that there is a unique sequence of columns visited by any path of length n, namely the intended one, by Lemma 1.

We only need to show that $t \geq \lfloor n/2 \rfloor$ for each n to apply Lemma 1.

It suffices to consider the worst case of n such that $a_k = 2^{2^{n-k}} - 1$, since for other values of n we have $t(n) = t(n-1) + 1$.

The long cycle will go from column 0 to column t where the length $k - t + 1$ of the sequence $2^t, \ldots, 2^k$ equals $n - k - 1$. Then we have $k - t + 1 = n - k - 1$ or $t = 2k - n + 2$. Then the claim is that $t \geq \lfloor n/2 \rfloor$.

To prove this, since we are in the worst case, we have $2^{2^{n-k}} - 1 = 2^k - 1$, i.e., $n = k + \log_2 k$. So we want to show

$$t = 2k - (k + \log_2 k) + 2 = 2 + k - \log_2 k \geq \lfloor n/2 \rfloor.$$

Since $2 + k - \log_2 k$ is an integer, it suffices to show

$$2 + k - \log_2 k \geq n/2 - \frac{1}{2},$$

which simplifies to

$$5 + k \geq 3 \log_2 k$$

which holds for all $k \geq 1$. □

Corollary 11. *The minimum number of states q such that $VC^{\text{upper}}_{n,q} = 2^n$ is at most $\sum_{i=0}^{k} a_i$ where $a_i = \min(2^i, 2^{2^{n-i}} - 1)$ and k is greatest such that $a_{k-1} < a_k$.*

Proof. Let $S \subseteq \{0,1\}^n$ be a set of words of length n. To accept all set words in a set F and reject all the words in $S \setminus F$, it certainly suffices to accept all the words in F, and reject all the words in $\{0,1\}^n \setminus F$, which is what Theorem 10 lets us do. □

Remark 2. Figure 4 shows an automaton such that among the words of length 4, it accepts exactly the ones with maximal nondeterministic automatic complexity. For reference, they are:

$$\{0010, 0011, 0100, 0110,$$
$$1101, 1100, 1011, 1001\}.$$

This information is available online at [6]. We use round brackets () for prefixes allowed and curly brackets {} for suffixes allowed. For instance, $(01), \{0\}$ means that we are in that state after reading $x_1 x_2 = 01$, and we are there again if $x_4 = 0$ will lead to accept. Dashed lines are for reading 0 and solid lines for reading 1.

In Theorems 12, 13, and 14 we collect some bounds on small VC-dimensions.

Theorem 12. $VC^{\text{upper}}_{q=6,n=4} \geq 12$.

Proof. If $\mathcal{F}$ for $n = 4$ does not contain any word starting with a certain length-2 prefix, say 11, then we can make do with just 6 states. And then we can still have $16 - 4 = 12$ words. □

Theorem 13. $VC^{\text{upper}}_{q=14,n=5} \geq 24$.

Proof. For $n = 5$, if one prefix of length 3 is absent from $\mathcal{F}$ then we can use one fewer states, namely 14, and still shatter a set of size $32 - 8 = 24$. □

We remark that Theorem 10 is not sharp. For instance, 4 states suffice for $n = 3$:

Theorem 14. *The length-restricted nondeterministic state complexity $s'_N(F)$ of a set of binary words of length $n = 3$ is at most 4.*

Proof. Use states corresponding to prefixes λ (the empty word), 0, 1, 00 and treat the latter 3 as the penultimate states in Champarnaud–Pin style:

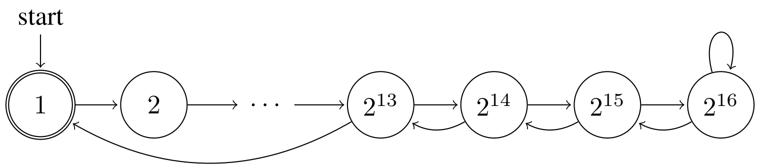
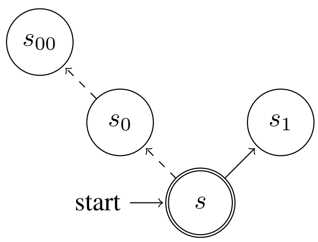


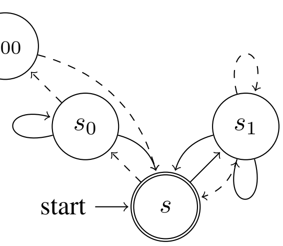
Figure 2: Automaton schema for Theorem 10.



(Here dashed lines indicate reading a 0 whereas solid lines indicate reading a 1.) For example, suppose we want to accept all words of length 3 except 001 and 010. This partly dictates how we proceed. The state s_{00} must correspond to the acceptance condition "accept only if the next bit is a 0". We can let states correspond to acceptance conditions as follows:

- s_0: accept only if the last bit is 1.
- s_1: accept no matter what the last bit is.
- s_{00}: accept only if the last bit is 0.

(We could also have switched the acceptance conditions for s_0 and s_1.) Then we can form the following automaton:



□

We can do better than Theorem 14: a computation determined [8] that 3 states suffice, although the pattern may be less uniform.

References

[1] Cezar Câmpeanu and Wing Hong Ho. The maximum state complexity for finite languages. *J. Autom. Lang. Comb.*, 9(2-3):189–202, 2004.

[2] J.-M. Champarnaud and J.-E. Pin. A maxmin problem on finite automata. *Discrete Applied Mathematics*, 23(1):91 – 96, 1989.

[3] Erik D. Demaine, Sarah Eisenstat, Jeffrey Shallit, and David A. Wilson. Remarks on separating words. In *Descriptional complexity of formal systems*, volume 6808 of *Lecture Notes in Comput. Sci.*, pages 147–157. Springer, Heidelberg, 2011.

[4] Kayleigh K. Hyde and Bjørn Kjos-Hanssen. Nondeterministic automatic complexity of overlap-free and almost square-free words. *Electron. J. Combin.*, 22(3):Paper 3.22, 18, 2015.

[5] Yoshiyasu Ishigami and Sei'ichi Tani. VC-dimensions of finite automata and commutative finite automata with k letters and n states. *Discrete Appl. Math.*, 74(3):229–240, 1997.

[6] Bjørn Kjos-Hanssen. Complexity lookup. http://math.hawaii.edu/wordpress/bjoern/complexity-of-0110100110010110/.

[7] Bjørn Kjos-Hanssen and Lei Liu. The number of languages with maximum state complexity. *Algebra Universalis*, 2020.

[8] Bjørn Kjos-Hanssen and Davin K. Takahashi. Code for "VC-dimensions of finite automata for words of equal length". https://github.com/bjoernkjoshanssen/vc, 2019.

[9] J. M. Robson. Separating strings with small automata. *Inform. Process. Lett.*, 30(4):209–214, 1989.

[10] Jeffrey Shallit. The separating words problem. https://cs.uwaterloo.ca/~shallit/Talks/sep2.pdf.

[11] Jeffrey Shallit and Ming-Wei Wang. Automatic complexity of strings. volume 6, pages 537–554. 2001. 2nd Workshop on Descriptional Complexity of Automata, Grammars and Related Structures (London, ON, 2000).

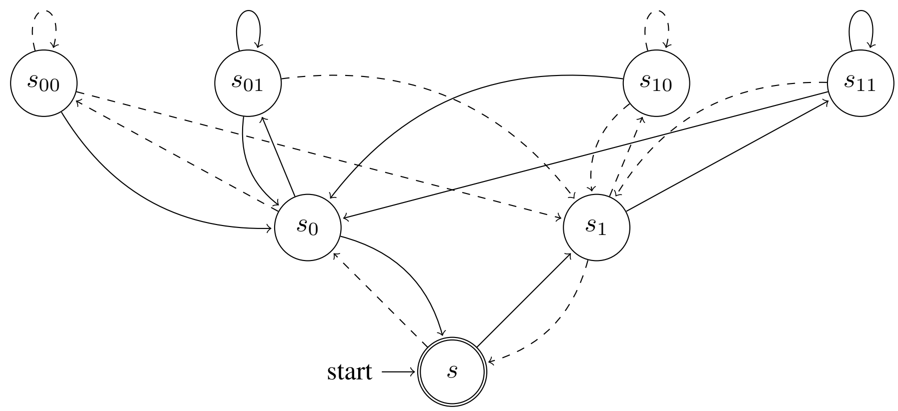


Figure 3: An automaton for $\{x_1 \ldots x_5 : x_2 = x_3, x_4 \neq x_5\}$. Note that s_0 has a dual function: in the beginning to record that $x_1 = 0$ and in the end to record that $x_4 = 1$. Dashed edges are labeled 0 and ordinary edges are labeled 1. This automaton saves more states than the general method of Theorem 10 guarantees, however.

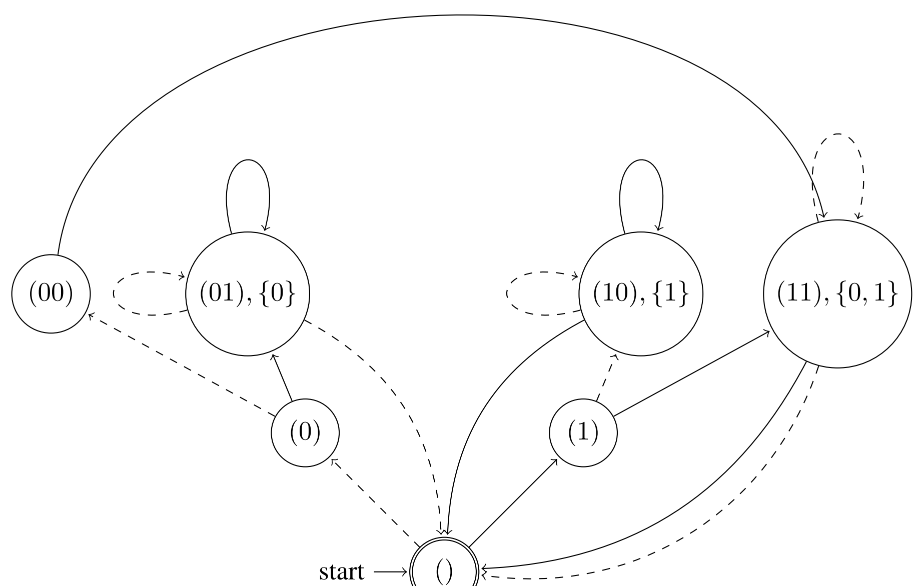


Figure 4: The $n = 4$ example from Remark 2.