Complexity of Infimal Observable Superlanguages

Tomáš Masopust

***Abstract*—The infimal prefix-closed, controllable and observable superlanguage plays an essential role in the relationship between controllability, observability and co-observability – the central notions of supervisory control theory. Existing algorithms for its computation are exponential and it is not known whether a polynomial algorithm exists. In this paper, we study the state complexity of this language. State complexity of a language is the number of states of the minimal DFA for the language. For a language of state complexity n, we show that the upper-bound state complexity on the infimal prefix-closed and observable superlanguage is $2^n + 1$ and that this bound is asymptotically tight. It proves that there is no algorithm computing a DFA of the infimal prefix-closed and observable superlanguage in polynomial time. Our construction further shows that such a DFA can be computed in time $O(2^n)$. The construction involves NFAs and a computation of the supremal prefix-closed sublanguage. We study the computation of the supremal prefix-closed sublanguage and show that there is no polynomial-time algorithm that computes an NFA of the supremal prefix-closed sublanguage of a language given as an NFA even if the language is unary.**

***Index Terms*—Discrete event systems; Automata; Prefix-closed language; Observable language; Complexity.**

I. Introduction

CONTROLLABILITY and observability are the central notions of supervisory control theory of discrete event systems in the Ramadge-Wonham framework [1]–[3]. They form the necessary and sufficient conditions for the existence of a supervisor that achieves the desired control behavior of a system. In decentralized supervisory control, where more supervisors cooperate to control the system, every supervisor observes and controls part of the system. The observation of a supervisor is modeled by an observation mask or by a natural projection. Cieslak et al. [1] and Rudie and Wonham [4] have shown that controllability and co-observability are the central notions in decentralized supervisory control.

A relationship between controllability, observability and co-observability has been studied by Kumar and Shayman [5], who have shown that the infimal prefix-closed, controllable and observable superlanguage plays the essential role. Another motivation and the importance of infimal superlanguages have been discussed in the fundamental book on supervisory control theory [6]. We have further illustrated its relevance to decentralized supervisory control with communication [7] and to coordination control [8]. We refer the reader to these papers for more details and examples.

Infimal superlanguages are of a general interest in supervisory control. There are examples in modular and decentralized control showing evidence that supremal sublanguages do not always suffice to achieve the best (optimal) solution and that the optimal solution may be achieved if infimal superlanguages are involved. The examples show evidence that the combination of supremal sublanguages and infimal superlanguages help achieve optimality if it is not achievable by supremal sublanguages alone [7], [8]. Therefore our interest in infimal prefix-closed, controllable and observable superlanguages.

Lafortune and Chen [9] have shown that the infimal prefix-closed and controllable superlanguage can be computed from a deterministic finite automaton (DFA) for the language in linear time. Kumar and Shayman [5] have further shown that it is sufficient to consider the computation of the infimal prefix-closed and observable superlanguage of a language K over Σ wrt the language Σ^*. Thus, we focus in this paper on the infimal prefix-closed and observable superlanguage of K wrt Σ^* and study its state complexity.

State complexity of a language is the number of states of the minimal DFA marking (accepting) the language. Since the minimal DFA is unique (up to isomorphism), state complexity is a complexity measure that is independent of the representation and computation of the language.

Our contribution: For a language K of state complexity n, we show that the upper-bound on the state complexity of the infimal prefix-closed and observable superlanguage of K wrt the language Σ^* is 2^n+1. We further prove that this bound is asymptotically tight by showing that the worst-case lower-bound state complexity is at least $\frac{3}{4} \cdot 2^n - 1 = \Omega(2^n)$. Since the state complexity is exponential, so is the time complexity of any algorithm computing the corresponding minimal DFA.

In addition, our construction shows that a DFA representation of the infimal prefix-closed and observable superlanguage of K wrt the language Σ^* can be computed in time $O(2^n)$.

Our construction involves nondeterministic finite automata (NFAs) and is based on a formula equivalent to the formulae of Rudie and Wonham [10] and of Kumar and Shayman [5]. The formulae include a computation of the supremal prefix-closed sublanguage. We study the computation of the supremal prefix-closed sublanguage and show that there is no polynomial-time algorithm computing an NFA representation of the supremal prefix-closed sublanguage of a language given as an NFA even if the language is unary.

II. Preliminaries

We assume that the reader is familiar with supervisory control theory [6] and automata theory [11], [12]. For undefined notions, the reader is refer to these references.

The prefix closure of a language L is the set $\overline{L} = \{w \in \Sigma^* \mid \text{there is } u \in \Sigma^* \text{ s.t. } wu \in L\}$; L is *prefix-closed* if $L = \overline{L}$. The *right quotient* of a language L wrt a language M is the set $L/M = \{w \in \Sigma^* \mid \text{there is } x \in M \text{ s.t. } wx \in L\}$. If $M = \{a\}$ is a singleton, we simply write $L/a = \{w \in \Sigma^* \mid wa \in L\}$. The *empty string* is denoted by ε.

Supported by the DFG in Emmy Noether grant KR 4381/1-1 (DIAMOND).

T. Masopust (masopust@math.cas.cz) is with CFEAD, TU Dresden, Germany, and with Institute of Mathematics, Czech Academy of Sciences

A *nondeterministic finite automaton* (NFA) is a quintuple $\mathcal{A} = (Q, \Sigma, \delta, Q_0, F)$, where Q is a finite nonempty set of states, Σ is an input alphabet, $Q_0 \subseteq Q$ is a set of initial states, $F \subseteq Q$ is a set of marked states, and $\delta\colon Q \times (\Sigma \cup \{\varepsilon\}) \to 2^Q$ is a transition function that is extended to $2^Q \times \Sigma^*$ by induction. The language *generated* by $\mathcal{A}$ is the set $L(\mathcal{A}) = \{w \in \Sigma^* \mid \delta(Q_0, w) \neq \emptyset\}$ and the language *marked* by $\mathcal{A}$ is the set $L_m(\mathcal{A}) = \{w \in \Sigma^* \mid \delta(Q_0, w) \cap F \neq \emptyset\}$.

The NFA $\mathcal{A}$ is an (incomplete) *deterministic finite automaton* (DFA) if $|Q_0| \leq 1$ and $|\delta(q, a)| \leq 1$ for every $q \in Q$ and $a \in \Sigma$. Moreover, DFAs do not admit ε-transitions, that is, δ is a partial transition function from $Q \times \Sigma$ to Q.

For every NFA $\mathcal{A}$ there exists a DFA $\mathcal{B}$ such that $L_m(\mathcal{B}) = L_m(\mathcal{A})$ and $L(\mathcal{B}) = L(\mathcal{A})$. The DFA $\mathcal{B}$ is constructed by the standard *subset construction* [12] and is called the *subset automaton* of $\mathcal{A}$. Specifically, for $\mathcal{A} = (Q, \Sigma, \delta, Q_0, F)$, $\mathcal{B} = (2^Q, \Sigma, \delta', Q_0, F')$, where $\delta'\colon 2^Q \times \Sigma \to 2^Q$ is defined as $\delta'(X, a) = \delta(X, a)$ and $F' = \{R \subseteq Q \mid R \cap F \neq \emptyset\}$.

Let Σ and Δ be alphabets. An *(observation) mask* is a map $P\colon \Sigma \to \Delta \cup \{\varepsilon\}$ that is extended to Σ^* so that $P(\varepsilon) = \varepsilon$ and $P(sa) = P(s)P(a)$ for $s \in \Sigma^*$ and $a \in \Sigma$. If L is a regular language, then $P(L) = \cup_{w\in L} P(w)$ is regular [13]. A mask P is a *(natural) projection* if $\Delta \subseteq \Sigma$ and $P(a) = a$, for $a \in \Delta$, and $P(a) = \varepsilon$ otherwise. The *inverse image* of a mask P, denoted by $P^{-1}\colon 2^{\Delta^*} \to 2^{\Sigma^*}$, is defined as $P^{-1}(L) = \{w \in \Sigma^* \mid P(w) \in L\}$. Regular languages are closed under the inverse image of a mask [13].

In the rest, the term *language* stands for a regular language.

III. Known and Preliminary Results

Let $\overline{\inf}\,\mathrm{CO}(K, L(G), \Sigma_u, P)$ denote the infimal superlanguage of K that is prefix-closed, controllable and observable wrt $L(G)$, uncontrollable events Σ_u, and a mask P. Similarly we use $\overline{\inf}\,\mathrm{C}(K, L(G), \Sigma_u)$ to denote the infimal prefix-closed and controllable superlanguage and $\overline{\inf}\,\mathrm{O}(K, L(G), P)$ to denote the infimal prefix-closed and observable superlanguage.

Kumar and Shayman [5] have proved that the computation of the infimal prefix-closed, controllable and observable superlanguage of K wrt $L(G)$ depends on the computation wrt Σ^*, namely $\overline{\inf}\,\mathrm{CO}(K, L(G), \Sigma_u, P) = \overline{\inf}\,\mathrm{CO}(K, \Sigma^*, \Sigma_u, P) \cap L(G)$. It thus suffices to consider the computation wrt the language Σ^*. They further proved that $\overline{\inf}\,\mathrm{CO}(K, \Sigma^*, \Sigma_u, P) = \overline{\inf}\,\mathrm{O}(\overline{\inf}\,\mathrm{C}(K, \Sigma^*, \Sigma_u), \Sigma^*, P)$. Lafortune and Chen [9] have shown that $\overline{\inf}\,\mathrm{C}(K, \Sigma^*, \Sigma_u) = \overline{K}\Sigma_u^*$, which can be computed from a DFA for K in linear time. The computation of the infimal prefix-closed and controllable superlanguage is thus easy and we focus in the rest on the computation of the infimal prefix-closed and observable superlanguage.

Rudie and Wonham [10] showed that $\overline{\inf}\,\mathrm{O}(K, L(G), P) = L(G) \setminus (\Sigma^+ \setminus \widetilde{P}^{-1}(\widetilde{P}(\overline{K})))\Sigma^*$, where P is a projection and $\widetilde{P}$ projects all but the last event, inductively defined by $\widetilde{P}(\varepsilon) = \varepsilon$ and $\widetilde{P}(sa) = P(s)a$. They also proved that for $K \neq \emptyset$,

$$\widetilde{P}^{-1}\widetilde{P}(\overline{K}) = \bigcup_{a\in\Sigma} \left[P^{-1}(P(\overline{K}a \cap \overline{K})) \cap \Sigma^*a\right] \cup \{\varepsilon\}\,. \tag{1}$$

The equation remains valid for masks and Kumar and Shayman [5] extended it and simplified to the form

$$\overline{\inf}\,\mathrm{O}(K, L(G), P) = \overline{\sup}\,[\widetilde{P}^{-1}\widetilde{P}(\overline{K})] \cap L(G) \tag{2}$$

where $\overline{\sup}\,(H)$ stands for the supremal prefix-closed sublanguage of a language H. Note that it immediately implies that $\overline{\inf}\,\mathrm{O}(K, L(G), P) = \overline{\inf}\,\mathrm{O}(K, \Sigma^*, P) \cap L(G)$.

The formulae consist of operations studied in the literature and their worst-case state complexities give a rough estimate on the state complexity of the language $\overline{\inf}\,\mathrm{O}(K, \Sigma^*, P)$. By Yu et al. [14], the bound is no more than $2^{|\Sigma|(4n^2+8n+1)}$, where n is the state complexity of K. Namely, Yu et al. [14] show that $\overline{K}a$ needs no more than $4n + 8$ states and $\overline{K}a \cap \overline{K}$ no more than $(4n + 8)n$ states. Then $P^{-1}P(\overline{K}a \cap \overline{K})$ needs at most $2^{(4n+8)n}$ states. (If P is a natural projection, the bound is lower [15], [16].) The intersection with Σ^*a then needs no more than $2^{(4n+8)n} \cdot 2$ states and the union over all events a in Σ no more than $(2^{(4n+8)n} \cdot 2)^{|\Sigma|}$ states. The supremal prefix-closed sublanguage of a DFA can be computed in linear time and does not increase the state complexity; it requires to remove all non-marked states and corresponding transitions.

Results of Yu et al. [14] hold for any language and the reader may notice that the languages of the formulae are of special forms. The worst-case state complexity of Yu et al. [14] is thus mostly not tight for them. For instance, it can be shown that the tight state complexity on $\overline{K}a \cap \overline{K}$ is $2n$ rather than $(4n + 8)n$, which decreases the upper bound to $2^{|\Sigma|2n}$.

We now show that the upper bound on the state complexity of the language $\overline{\inf}\,\mathrm{O}(K, \Sigma^*, P)$ is no more than $2^n + 1$. To this aim, we express the formula for $\overline{\inf}\,\mathrm{O}(K, \Sigma^*, P)$ in an equivalent form using the operation of right quotient. This expression is based on the following relation between the mask, intersection and right quotient operations.

Lemma 1: Let P be a mask from Σ to Δ. For a prefix-closed language K over Σ and an event $a \in \Sigma$, it holds that

$$P^{-1}(P(Ka \cap K)) \cap \Sigma^*a = (P^{-1}P(K/a))a\,.$$

Proof: The claim holds for $K = \emptyset$. Assume that $K \neq \emptyset$. Let $xa \in P^{-1}(P(Ka \cap K)) \cap \Sigma^*a$. Then $P(xa) \in P(Ka \cap K)$ and there exists $ya \in Ka \cap K$ such that $P(xa) = P(ya)$. Since $ya \in K$, we have that $y \in K/a$, hence $xa \in P^{-1}(P(y))a \subseteq (P^{-1}P(K/a))a$. On the other hand, let $xa \in (P^{-1}P(K/a))a$. Then $x \in P^{-1}P(K/a)$ and there is $y \in K/a$ with $P(x) = P(y)$. Since $y \in K/a$, $ya \in K$. Because K is prefix-closed, $y \in K$, which implies that $ya \in Ka \cap K$. Thus, $P(xa) \in P(Ka \cap K)$, that is, $xa \in P^{-1}(P(Ka \cap K)) \cap \Sigma^*a$. ■

The assumption that the language is prefix-closed is essential. The lemma does not hold for non-prefix-closed languages even if P is the identity mask. In this case, Lemma 1 reduces to $Ka \cap K = (K/a)a$. If $K = \{aa\}$ is non-prefix-closed, then $Ka \cap K = \emptyset$, whereas $(K/a)a = \{aa\}$.

We can now express the formula of Kumar and Shayman [5] in an equivalent form using the operation of right quotient.

Theorem 2: Let K be a nonempty language over Σ, and let P be a mask from Σ to Δ. Then $\overline{\inf}\,\mathrm{O}(K, \Sigma^*, P) = \overline{\sup}\,(\cup_{a\in\Sigma}(P^{-1}P(\overline{K}/a))a \cup \{\varepsilon\})$.

Proof: By (1), (2), and Lemma 1, $\overline{\inf}\,\mathrm{O}(K, \Sigma^*, P) = \overline{\sup}\,(\widetilde{P}^{-1}\widetilde{P}(\overline{K})) = \overline{\sup}\,(\cup_{a\in\Sigma}\,[P^{-1}(P(\overline{K}a \cap \overline{K})) \cap \Sigma^*a] \cup \{\varepsilon\}) = \overline{\sup}\,(\cup_{a\in\Sigma}\,(P^{-1}P(\overline{K}/a))a \cup \{\varepsilon\})$, respectively. ■

We further modify the formula by moving the union operation deeper into the formula. It is then applied to a structurally simpler subformula, which is useful for our goal.

Lemma 3: Let $K \subseteq \Sigma^*$ be a language and $P\colon \Sigma \to \Delta \cup \{\varepsilon\}$ be a mask. Let $\Sigma' = \{a' \mid a \in \Sigma\}$ be a copy of Σ disjoint from both Σ and Δ. Let $h\colon \Sigma \cup \Sigma' \to \Delta \cup \Sigma' \cup \{\varepsilon\}$ be a mask defined by $h(a) = P(a)$, for $a \in \Sigma$, and $h(a') = a'$, for $a' \in \Sigma'$. Let $g\colon \Sigma' \to \Sigma$ be a mask defined by $g(a') = a$, for $a' \in \Sigma'$. Then

$$\bigcup_{a\in\Sigma} (P^{-1}P(K/a))a = g\left(h^{-1}h\left(\bigcup_{a\in\Sigma}(K/a)a'\right) \cap \Sigma^*\Sigma'\right) .$$

Proof: By the properties of masks, we have that

$$\begin{aligned}
&g(h^{-1}(h(\cup_{a\in\Sigma}(K/a)a')) \cap \Sigma^*\Sigma') \\
&= g([\cup_{a\in\Sigma}\ h^{-1}(h((K/a)a'))] \cap \Sigma^*\Sigma') \\
&= g([\cup_{a\in\Sigma}\ h^{-1}(h(K/a)h(a'))] \cap \Sigma^*\Sigma') \\
&= g([\cup_{a\in\Sigma}\ h^{-1}(P(K/a)a')] \cap \Sigma^*\Sigma') \\
&= g([\cup_{a\in\Sigma}\ h^{-1}(P(K/a))h^{-1}(a')] \cap \Sigma^*\Sigma') \\
&= g([\cup_{a\in\Sigma}\ P^{-1}(P(K/a))a'P^{-1}(\varepsilon)] \cap \Sigma^*\Sigma') \\
&= g(\cup_{a\in\Sigma}\ [P^{-1}(P(K/a))a'P^{-1}(\varepsilon) \cap \Sigma^*\Sigma']) \\
&= g(\cup_{a\in\Sigma}\ P^{-1}(P(K/a))a') \\
&= \cup_{a\in\Sigma}\ g(P^{-1}(P(K/a))a') \\
&= \cup_{a\in\Sigma}\ (P^{-1}P(K/a))a\,.
\end{aligned}$$

This completes the proof. ■

As a corollary of Theorem 2 and Lemma 3, we obtain the following formula, which we use to show the asymptotically tight bound on the state complexity of $\overline{\inf}\,\mathrm{O}(K, \Sigma^*, P)$.

Corollary 4: Under the assumptions of Lemma 3, if $K \neq \emptyset$,

$$\overline{\inf}\,\mathrm{O}(K, \Sigma^*, P) = \overline{\sup}\left[g\left(h^{-1}h\left(\bigcup_{a\in\Sigma}(\overline{K}/a)a'\right) \cap \Sigma^*\Sigma'\right) \cup \{\varepsilon\}\right] .$$

IV. Deterministic State Complexity

We now use Corollary 4 to show that $2^n + 1$ is an upper-bound on the state complexity of the language $\overline{\inf}\,\mathrm{O}(K, \Sigma^*, P)$ and that the bound is asymptotically tight.

Corollary 4 suggests an algorithm (Algorithm 1) to compute the language $\overline{\inf}\,\mathrm{O}(K, \Sigma^*, P)$. We now discuss state complexities of its steps. Consequently we obtain its time complexity.

Lemma 5 (Yu et al. [14]): Let $\mathcal{A}$ be a DFA over Σ with n states, and let $a \in \Sigma$. Then the minimal DFA for $L_m(\mathcal{A})/a$ has at most n states. The bound is tight.

```
Algorithm 1 Computation of $\overline{\inf}\,\mathrm{O}(K, \Sigma^*, P)$
Input: a DFA for $K$ over $\Sigma$ and a mask $P$
Output: a DFA for the language $\overline{\inf}\,\mathrm{O}(K, \Sigma^*, P)$
1: if $K = \emptyset$ then return the DFA for $K$
2: else
3:     Compute a DFA for $\overline{K}$
4:     Compute a DFA for $\cup_{a\in\Sigma}(\overline{K}/a)a'$
5:     Compute an NFA for $g(h^{-1}h(\cup_{a\in\Sigma}(\overline{K}/a)a') \cap \Sigma^*\Sigma')$
6:     Determinize the NFA
7:     Compute the union with $\{\varepsilon\}$
8:     Compute the supremal prefix-closed sublanguage
```

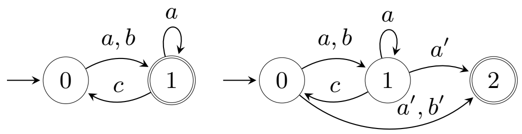

Fig. 1. Automata $\mathcal{A}$ (left) and $\mathcal{B}$ (right) for $\cup_{a\in\Sigma}(L_m(\mathcal{A})/a)a'$

The construction is as follows. Let $\mathcal{A} = (Q, \Sigma, \delta_\mathcal{A}, q_0, F_\mathcal{A})$ be a DFA. Construct the DFA $\mathcal{A}' = (Q, \Sigma, \delta_\mathcal{A}, q_0, F_{\mathcal{A}'})$, where $F_{\mathcal{A}'} = \{q \in Q \mid \delta_\mathcal{A}(q, a) \in F_\mathcal{A}\}$. Then $L_m(\mathcal{A}') = L_m(\mathcal{A})/a$.

We now study the size of the minimal DFA for the language computed in Step 4 of the algorithm.

Lemma 6: Let $\mathcal{A}$ be a DFA over Σ with n states. Then the minimal DFA for $\cup_{a\in\Sigma}\,(L_m(\mathcal{A})/a)a'$ has at most $n+1$ states. The bound is tight even for prefix-closed languages.

Proof: Let $\mathcal{A} = (Q, \Sigma, \delta_\mathcal{A}, q_0, F_\mathcal{A})$ be a DFA with n states $Q = \{0, 1, \ldots, n-1\}$. For every $a \in \Sigma$, we construct the set $F_a = \{q \in Q \mid \delta_\mathcal{A}(q, a) \in F_\mathcal{A}\}$ of all states of $\mathcal{A}$ from which an a-transition reaches a marked state. We construct the DFA $\mathcal{B} = (Q \cup \{n\}, \Sigma, \delta_\mathcal{B}, 0, \{n\})$ from $\mathcal{A}$ by adding a new state, n, which is the only marked state, and by defining the transitions $\delta_\mathcal{B}(q, a) = \delta_\mathcal{A}(q, a)$, for $0 \le q \le n-1$ and $a \in \Sigma$, and $\delta_\mathcal{B}(f, a') = n$, for every $f \in F_a$. The construction is illustrated in Fig. 1. The corresponding sets are $F_a = \{0, 1\}$, $F_b = \{0\}$ and $F_c = \emptyset$.

We claim that $\mathcal{B}$ marks the language $\cup_{a\in\Sigma}\,(L_m(\mathcal{A})/a)a'$. If a string is marked by $\mathcal{B}$, it is of the form wa', for some $a \in \Sigma$, which means that $\delta_\mathcal{B}(0, w) \in F_a$. By the construction of F_a, $w \in L_m(\mathcal{A})/a$, hence $wa' \in (L_m(\mathcal{A})/a)a'$. On the other hand, if $wa' \in \cup_{a\in\Sigma}\,(L_m(\mathcal{A})/a)a'$, then $w \in L_m(\mathcal{A})/a$, hence $\delta_\mathcal{B}(0, w) = f_a$, for some $f_a \in F_a$, which implies that $\delta_\mathcal{B}(0, wa') = \delta_\mathcal{B}(f_a, a') = n$, hence it is marked by $\mathcal{B}$.

To show that the bound is tight, we consider the DFA $\mathcal{A}$ depicted in Fig. 2 (solid arrows) with states $\{0, \ldots, n-1\}$, where state 0 is initial and all states are marked. The DFA is minimal; two states are distinguishable by a string in b^*. The DFA $\mathcal{B}$ for $(L_m(\mathcal{A})/a)a' \cup (L_m(\mathcal{A})/b)b'$ is depicted in Fig. 2 (all arrows), where the states are $\{0, \ldots, n\}$ with n being the only marked state. There is an a'-transition from state i to state n for every $0 \le i \le n-1$, and a b'-transition from state j to state n for every $0 \le j \le n-2$. The DFA $\mathcal{B}$ is minimal; states $\{0, \ldots, n-1\}$ are distinguishable by the same argument as for $\mathcal{A}$ and n is not equivalent with any other state since it is the only marked state. ■

We now use the previous results to obtain our upper-bound on the state complexity of the language $\overline{\inf}\,\mathrm{O}(K, \Sigma^*, P)$.

Theorem 7 (Upper bound): Let K over Σ be a nonempty language marked by a DFA with n states. Then the minimal

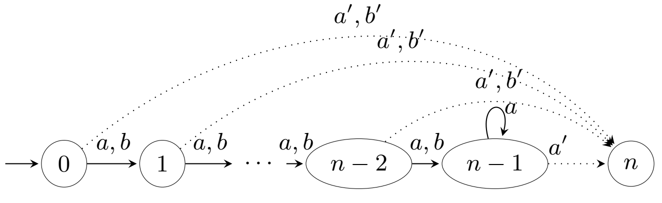

Fig. 2. Automata $\mathcal{A}$ (solid arrows) and $\mathcal{B}$ (all arrows)

DFA for $\overline{\inf}\,\mathrm{O}(K,\Sigma^*,P)$ has no more than 2^n+1 states.

Proof: Let $P\colon \Sigma \to \Delta \cup \{\varepsilon\}$. By Corollary 4, we have that $\overline{\inf}\,\mathrm{O}(K,\Sigma^*,P) = \overline{\sup}\,[g(h^{-1}h(\cup_{a\in\Sigma}(\overline{K}/a)a') \cap \Sigma^*\Sigma') \cup \{\varepsilon\}]$. From Lemma 6, we have that the minimal DFA marking the language $\cup_{a\in\Sigma}(\overline{K}/a)a'$ has at most $n+1$ states, only one of which is marked. We denote this state by f. Notice that, by the construction, there is no transition from state f.

We represent the language $g(h^{-1}h(\cup_{a\in\Sigma}(\overline{K}/a)a')\cap\Sigma^*\Sigma')$ as an NFA as follows. The language $h(\cup_{a\in\Sigma}(\overline{K}/a)a')$ is computed by replacing every x-transition, $x\in\Sigma$, with the $h(x)$-transition. The language $h^{-1}h(\cup_{a\in\Sigma}(\overline{K}/a)a')$ is then computed by replacing every y-transition, $y\in\Delta$, by an x-transition for every $x\in\Sigma$ such that $h(x)=y$. In addition, for every $x\in\Sigma$ such that $h(x)=\varepsilon$, we add a self-loop under x to every state of the NFA. To compute an NFA for $h^{-1}h(\cup_{a\in\Sigma}(\overline{K}/a)a')\cap\Sigma^*\Sigma'$ then means to remove all transitions from state f. This can be done during the computation of an NFA for $h^{-1}h(\cup_{a\in\Sigma}(\overline{K}/a)a')$ so that no self-loop is added to state f. The computation of an NFA for the mask g is similar to that of h.

The resulting NFA has at most $n+1$ states. Thus, a DFA equivalent to the NFA, constructed by the standard subset construction, has at most 2^{n+1} reachable states. However, since every marked state of the subset automaton must contain f, and there are at most 2^n subsets containing f, there are at most 2^n marked states in the computed DFA.

To compute the union with $\{\varepsilon\}$, the DFA may require one more (initial and marked) state. Thus, the resulting DFA has at most $2^{n+1}+1$ states, where at most 2^n+1 states are marked.

Since $\overline{\inf}\,\mathrm{O}(K,\Sigma^*,P)$ is prefix-closed, its minimal DFA must have all states marked. There are at most 2^n+1 marked states in the above constructed automaton, therefore the minimal DFA for $\overline{\inf}\,\mathrm{O}(K,\Sigma^*,P)$ can have at most so many states. ■

Consequently, the time complexity of Algorithm 1 is $O(2^n)$. Indeed, let n be the state complexity of K. Step 3 requires time $O(n)$. To compute Step 4, we add a new state, f, and scan the automaton in linear time using, e.g., the breadth-first search (BFS) algorithm [17]. For every state q and its out-going transition under x, if $\delta(q,x)$ is marked, we add an x'-transition from q to f. This can be done in time $O(1+n+2n\cdot|\Sigma|)=O(n\cdot|\Sigma|)$, since there are n states, one added new state, and at most $n\cdot|\Sigma|$ transitions that may be duplicated to f. Step 5 can be computed in time $O(n\cdot|\Sigma|)$ as follows. The application of h can be done in time $O(n\cdot|\Sigma|)$ by the BFS algorithm. The application of h^{-1} can be done in time $O(n\cdot|\Sigma|+n\cdot|\Sigma\setminus\Delta|)=O(n\cdot|\Sigma|)$, where the second part corresponds to adding self-loops under unobservable events. As explained above, the intersection with $\Sigma^*\Sigma'$ is done so that no transitions are added to f during the computation of h^{-1}. Step 6 can be computed in time $O(2^n\cdot|\Sigma|)$, since, by the proof of Theorem 7, the DFA has at most $2^{n+1}+1$ states and $|\Sigma|$ transitions in every state. Step 7 can be computed in time $O(|\Sigma|)$ as follows: let q_0 be the initial state of the DFA, and let q_i be a new marked state. We change the DFA so that q_i is the only initial state, i.e., q_0 is not initial anymore, and for every $x\in\Sigma$, we define $\delta(q_i,x)=\delta(q_0,x)$. Finally, Step 8 can be computed in linear time wrt the size of the input DFA by removing all non-marked states and the corresponding transitions. The overall time complexity is $O(|\Sigma|\cdot 2^n)$. Considering the size of the alphabet as constant results in the claimed complexity $O(2^n)$.

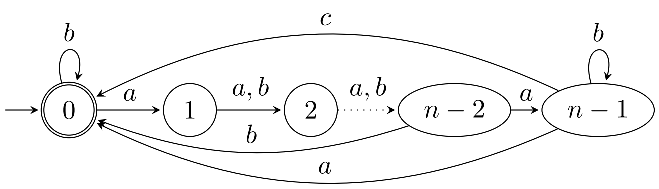

Fig. 3. The minimal DFA $\mathcal{A}_n$ for K_n

We now discuss the lower-bound state complexity and show that it is $\Omega(2^n)$. It holds even for projections.

Theorem 8 (Lower bound): Let $P\colon\{a,b,c\}^*\to\{a,b\}^*$ be a projection. For every $n\ge 2$, there exists a minimal DFA with n states marking a language $K_n\subseteq\{a,b,c\}^*$, such that the state complexity of $\overline{\inf}\,\mathrm{O}(K_n,\Sigma^*,P)$ is at least $\frac{3}{4}\cdot 2^n-1$.

Proof: Let K_n be the language marked by the DFA $\mathcal{A}_n$ depicted in Fig. 3. It has n states $\{0,1,\dots,n-1\}$, where state 0 is the sole initial and marked state. For $0\le i\le n-1$, $\delta(i,a)=(i+1 \bmod n)$. For $1\le i\le n-3$, $\delta(i,b)=i+1$, $\delta(n-2,b)=0$, and, for $i\in\{0,n-1\}$, $\delta(i,b)=i$. Finally, there is a single c-transition $\delta(n-1,c)=0$.

An NFA $\mathcal{B}_n$ for the language $g(h^{-1}h(\cup_{a\in\Sigma}(\overline{K_n}/a)a')\cap\Sigma^*\Sigma')$ is build from $\mathcal{A}_n$ according to the above constructions in the following steps and the result is depicted in Fig. 4:

1) We compute $\overline{K_n}$ by marking all states of $\mathcal{A}_n$.
2) To compute $\cup_{a\in\Sigma}(\overline{K_n}/a)a'$, we add a new state, n. From every state of $\mathcal{A}_n$, transitions under a' and b' go to state n, and a transition under c' goes from state $n-1$ to state n. The only marked state is state n.
3) The language $h(\cup_{a\in\Sigma}(\overline{K}/a)a')$ is computed by replacing the c-transition by an ε-transition.
4) To compute $h^{-1}h(\cup_{a\in\Sigma}(\overline{K}/a)a')\cap\Sigma^*\Sigma'$, a self-loop under c is added to every state of $\mathcal{A}_n$. Note that it is not added to state n, since it would be eliminated by the intersection with $\Sigma^*\Sigma'$. Thus, this can be done in linear time without computing the intersection.
5) Finally, to apply g means to rename all transitions under a', b' and c', which all go to state n.

We show that the minimal DFA equivalent to the NFA $\mathcal{B}_n$ has at least $\frac{3}{4}\cdot 2^n-1$ reachable marked states. Using the standard subset construction, we first show that all states of the

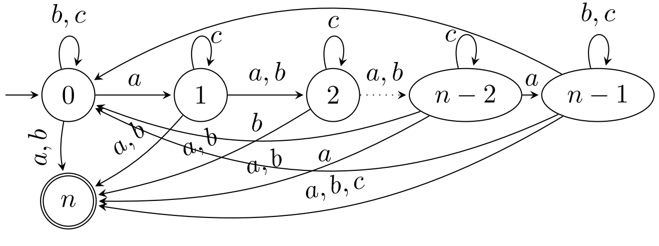

Fig. 4. An NFA $\mathcal{B}_n$ marking language $g(h^{-1}h\left(\bigcup_{a\in\Sigma}(\overline{K_n}/a)a'\right)\cap\Sigma^*\Sigma')$

subset automaton corresponding to the NFA $\mathcal{B}_n$ are pairwise distinguishable. Indeed, $\mathcal{B}_n$ marks ε only from state n and $a^i c$ only from state $n-1-i$, for $0 \leq i \leq n-1$. Therefore, the states of the subset automaton are pairwise distinguishable. To prove the theorem, we show that the subset automaton has $2^{n-1} + 2^{n-2} - 1$ marked states that are all reachable via other marked states.

State $\{0\}$ is initial, but not marked; we resolve this issue later. We now prove, by induction on the size of the subset, that every subset of $\{0, 1, \ldots, n-1, n\}$ containing 0 and n is reachable in the subset automaton from state $\{0\}$ by a nonempty string over $\{a, b\}$. Since there is an a-transition and a b-transition from every state 0 through $n-1$ to n, all subsets reachable by such a string must contain state n, i.e., they are marked in the subset automaton. State $\{0, n\}$ is reachable from state $\{0\}$ by b. State $\{n-2, n\}$ is reachable from $\{0\}$ by a^{n-2}. State $\{0, n-2, n\}$ is reachable from state $\{n-2, n\}$ by $a^2 b^{n-3}$. State $\{0, n-2, n\}$ goes to state $\{0, 1, n-1, n\}$ by a, and then by a string in b^* to states $\{0, i, n-1, n\}$ with $1 \leq i \leq n-2$. State $\{0, n-2, n-1, n\}$ goes to state $\{0, n-1, n\}$ by b, and then to state $\{0, 1, n\}$ by a. By a string in b^*, state $\{0, 1, n\}$ goes to states $\{0, i, n\}$ with $1 \leq i \leq n-2$. Thus, each subset of size two or three containing 0 and n is reachable.

Now, let $X = \{0, i_1, i_2, \ldots, i_t, n\}$ be a set of size $t+2$, where $2 \leq t \leq n-1$ and $1 \leq i_1 < i_2 < \cdots < i_t \leq n-1$. We consider two cases:

1) If $i_t = n-1$, then X is reachable from state $\{0, i_2 - i_1, \ldots, i_{t-1} - i_1, n-2, n\}$ by $ab^{i_1 - 1}$, and the latter set of size $t+1$ is reachable by the induction hypothesis.
2) If $i_t < n-1$, then X is reachable from state $\{0, i_2 - i_1, \ldots, i_t - i_1, n-1, n\}$ by $ab^{i_1 - 1}$, and the latter set of size $t+2$ contains state $n-1$, and is reachable by 1).

This proves reachability of all subsets of $\{0, 1, \ldots, n\}$ containing 0 and n. There are 2^{n-1} such subsets.

Next, if $X = \{i_1, i_2, \ldots, i_t\}$ is a non-empty subset of the set $\{1, 2, \ldots, n-2\}$, then the set $X \cup \{n\}$ is reachable from the set $\{0, i_2 - i_1, i_3 - i_1, \ldots, i_t - i_1, n\}$ containing 0 and n by a^{i_1}. Thus, for every $\emptyset \neq X \subseteq \{1, 2, \ldots, n-2\}$, state $X \cup \{n\}$ is reachable in the subset automaton. These sets do not contain 0, hence they are different from the reachable states considered above. There are $2^{n-2} - 1$ such subsets.

Finally, we compute the union with the language $\{\varepsilon\}$. To do this, we create a new initial and accepting state, I, (state $\{0\}$ is not initial anymore) with transitions defined exactly as for state $\{0\}$, that is, $\delta(I, x) = \delta(\{0\}, x)$, for every $x \in \{a, b, c\}$. This has resolved the problem with the non-marked initial state, since state I is marked and has the same transitions as state $\{0\}$, that is, all states reachable from state $\{0\}$ are also reachable from state I. Thus, we have shown that the minimal DFA constructed by the subset construction has at least $2^{n-1} + 2^{n-2}$ marked states that are all reachable from the initial marked state I via marked states.

However, state I is equivalent to state $\{0, n\}$. Indeed, both states I and $\{0, n\}$ go to state $\{1, n\}$ under a, to state $\{0, n\}$ under b, and to state $\{0\}$ under c.

It remains to show that if the non-marked states are eliminated, the constructed marked states different from I are still pairwise distinguishable. Let X and Y be two sets different from I constructed above. They both contain n and, without loss of generality, we may assume that there exists i such that $n-1-i \in X \setminus Y$. Then the set reachable from X under a^i contains $n-1$, but the set reachable from Y under a^i does not. It means that $a^i c$ is marked from X, but not from Y, which distinguishes the states X and Y. Therefore, the minimal DFA of the supremal prefix-closed sublanguage has at least $2^{n-1} + 2^{n-2} - 1$ states, which completes the proof. ■

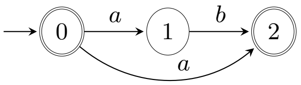

Fig. 5. A prefix-closed NFA $\mathcal{A}$ with $L_m(\mathcal{A}) = \overline{\{ab\}}$

Combining the upper and lower bounds of Theorems 7 and 8 gives the following corollary.

Corollary 9: Let K over Σ be a language with state complexity n, and let P be a mask. Then the worst-case state complexity of the language $\overline{\inf}\,\mathrm{O}(K, \Sigma^*, P)$ is $\Theta(2^n)$. ■

We also have the following consequence on the time complexity of Algorithm 1.

Corollary 10: The time complexity of Algorithm 1 is $\Theta(2^n)$, where n is the state complexity of the input language. ■

V. Nondeterministic State Complexity

Algorithm 1 represents the language as an NFA and it is determinized before computing the operation $\overline{\sup}\,(\cdot)$. The algorithm computing $\overline{\sup}\,(\cdot)$ on a DFA cuts off all non-marked states and the corresponding transitions, which requires linear time wrt the size of the input DFA. However, as shown above, this DFA may be exponentially larger than the DFA for K.

Another possibility is to execute $\overline{\sup}\,(\cdot)$ directly on an NFA. We now discuss this possibility and show that, in general, there is no polynomial-time algorithm that, given an NFA $\mathcal{A}$, would compute an NFA marking the language $\overline{\sup}\,(L_m(\mathcal{A}))$.

We first provide a brief insight into the difference between the computation of $\overline{\sup}\,(\cdot)$ for DFAs and NFAs. Indeed, if all states of an NFA are marked, then its language is prefix-closed. However, compared to DFAs, the problem with NFAs is that having a non-marked state does not yet mean that the language is not prefix-closed, cf. Fig. 5 for an example. It can be shown that, given an NFA, it is PSPACE-complete to decide whether its marked language is prefix-closed [18].

Theorem 11: The problem whether the marked language of an NFA is prefix-closed is PSPACE-complete. ■

We now show that there is no polynomial-time algorithm computing an NFA representation of $\overline{\sup}\,(L_m(\mathcal{A}))$ in general.

Theorem 12: Let $\mathcal{A}$ be an NFA. There is no polynomial-time algorithm computing an NFA for the language $\overline{\sup}\,(L_m(\mathcal{A}))$. The claim holds even for unary languages.

Proof: We prove the theorem by constructing, for any $n \geq 1$, an NFA $\mathcal{A}_n$ with polynomially many states in n such that any NFA for $\overline{\sup}\,(L_m(\mathcal{A}_n))$ has at least exponentially many states in n. Clearly, such an NFA cannot be computed in polynomial time wrt the size of $\mathcal{A}_n$.

To construct the NFAs $\mathcal{A}_n$, we first construct auxiliary DFAs $\mathcal{B}_n$, for every $n \geq 0$. The DFA $\mathcal{B}_0 = (X_0, \{a\},$

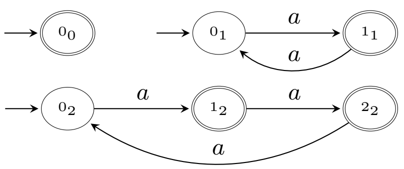

Fig. 6. The NFA $\mathcal{A}_2$; "nondeterministic" union of DFAs $\mathcal{B}_0$, $\mathcal{B}_1$, and $\mathcal{B}_2$

$\gamma_0, X_{i,0}, X_{m,0})$, where $X_0 = X_{i,0} = X_{m,0} = \{0_0\}$ and $\gamma_0(0_0, a)$ is undefined. For $n \geq 1$, let p_n denote the nth prime number. We define the DFA $\mathcal{B}_n = (X_n, \{a\}, \gamma_n, X_{i,n}, X_{m,n})$, where the state set is $X_n = \{0_n, 1_n, \ldots, (p_n - 1)_n\}$, the set of initial states is $X_{i,n} = \{0_n\}$, the set of marked states is $X_{m,n} = X_n \setminus \{0_n\}$, and the transition function is $\gamma_n(i_n, a) = (i + 1 \bmod p_n)_n$, for all $i_n \in X_n$. Then $L_m(\mathcal{B}_0) = \{\varepsilon\}$ and $L_m(\mathcal{B}_n) = a^* \setminus (a^{p_n})^*$, cf. Fig. 6 for automata $\mathcal{B}_0$, $\mathcal{B}_1$, and $\mathcal{B}_2$. We assume that the state sets X_i and X_j are disjoint for any $i \neq j$.

For $n \geq 1$, we build the NFA $\mathcal{A}_n = (Q_n, \{a\}, \delta_n, Q_{i,n}, F_n)$ as a "nondeterministic" union of the DFAs $\mathcal{B}_0, \mathcal{B}_1, \ldots, \mathcal{B}_n$. The NFA $\mathcal{A}_2$ is depicted in Fig. 6. Formally, $Q_n = \cup_{k=0}^{n} X_k$, $\delta_n(i_k, a) = \gamma_k(i_k, a)$, $Q_{i,n} = \cup_{k=0}^{n} X_{i,k}$, and $F_n = \cup_{k=0}^{n} X_{m,k}$. The number of states of $\mathcal{A}_n$ is $1 + \sum_{i=1}^{n} p_i$, which has been estimated by Bach and Shallit [19] to be $1 + 2^{-1}n^2 \ln n = O(n^2 \ln n)$. The marked language of $\mathcal{A}_n$ is $L_m(\mathcal{A}_n) = a^* \setminus (a^{p_n\#})^+$, where $p_n\# = \Pi_{i=1}^{n} p_i$. Indeed, for $m \geq 1$, string a^m is marked by $\mathcal{A}_n$ if and only if there is $p_i \in \{p_1, \ldots, p_n\}$ such that $m \bmod p_i \neq 0$. Thus, the shortest string that is not marked by $\mathcal{A}_n$ is of length $p_n\#$. Therefore, the supremal prefix-closed sublanguage of $L_m(\mathcal{A}_n)$ is the finite language $\overline{\{a^{p_n\#-1}\}}$.

We now show, using the fooling set technique [20], that any NFA marking this language requires at least $p_n\#$ states.

Fact 13 (Fooling set technique): Let $L \subseteq \Sigma^*$ be a language, and let $S = \{(x_i, y_i) \mid 1 \leq i \leq k\}$ be a set of pairs such that

(i) $x_i y_i \in L$ for $1 \leq i \leq k$, and

(ii) if $i \neq j$, then $x_i y_j \notin L$ or $x_j y_i \notin L$, for $1 \leq i, j \leq k$.

Then any NFA marking the language L has at least k states. Set S is called a fooling set for L. □

Let $S = \{(a^i, a^{p_n\#-i-1}) \mid 0 \leq i \leq p_n\# - 1\}$. Then $a^{i+p_n\#-i-1} = a^{p_n\#-1}$ belongs to the language $\overline{\{a^{p_n\#-1}\}}$. Thus, S satisfies item (i) of the fooling set technique. To show that it also satisfies item (ii), let $(a^i, a^{p_n\#-i-1})$ and $(a^j, a^{p_n\#-j-1})$ be two elements of S. Without loss of generality, we assume that $i < j$. Then $j + p_n\# - i - 1 > p_n\# - 1$, which implies that $a^j a^{p_n\#-i-1}$ does not belong to $\overline{\{a^{p_n\#-1}\}}$, i.e., it proves that S satisfies item (ii). Thus, S is a fooling set for the language $\overline{\{a^{p_n\#-1}\}}$ of size $p_n\#$. Therefore, any NFA marking the language $\overline{\{a^{p_n\#-1}\}}$ has at least $p_n\#$ states. Since $p_n\# = e^{(1+o(1))n \log n}$ [21] is exponential wrt n, hence not polynomial wrt the size of $\mathcal{A}_n$, there is no algorithm that would compute an NFA for the language $\overline{\{a^{p_n\#-1}\}}$ in polynomial time. ■

VI. Conclusion

A consequence of the exponential state complexity is that any algorithm computing a DFA for $\overline{\inf}\,\mathrm{O}(K, \Sigma^*, P)$ requires, in the worst case, exponential time (and exponential space to store it). Algorithm 1 further shows that the exponential time is sufficient. The algorithm is thus optimal in the sense that there is no asymptotically more efficient algorithm.

Concerning the NFA representation, we showed that even for unary languages, the algorithm would need more than polynomial time to compute the result and more than polynomial space to store it. This is in contrast to checking whether the language of an NFA is prefix closed, which can be done in polynomial space and it is not known whether it can be done in polynomial time.

Acknowledgment

The author gratefully acknowledges very useful suggestions and comments of the anonymous referees. The upper bound on the state complexity of the operation $\overline{K}a \cap \overline{K}$ is due to an anonymous referee.

References

[1] R. Cieslak, C. Desclaux, A. S. Fawaz, and P. Varaiya, "Supervisory control of discrete-event processes with partial observations," *IEEE Trans. Automat. Control*, vol. 33, pp. 249–260, 1988.

[2] F. Lin and W. M. Wonham, "On observability of discrete-event systems," *Inform. Sci.*, vol. 44, no. 3, pp. 173–198, 1988.

[3] P. J. Ramadge and W. M. Wonham, "The control of discrete event systems," *Proc. of the IEEE*, vol. 77, pp. 81–98, 1989.

[4] K. Rudie and W. M. Wonham, "Think globally, act locally: Decentralized supervisory control," *IEEE Trans. Automat. Control*, vol. 37, no. 11, pp. 1692–1708, 1992.

[5] R. Kumar and M. A. Shayman, "Formulae relating controllability, observability, and co-observability," *Automatica*, vol. 34, no. 2, pp. 211–215, 1998.

[6] C. G. Cassandras and S. Lafortune, *Introduction to discrete event systems*, 2nd ed. Springer, 2008.

[7] J. Komenda and T. Masopust, "Computation of controllable and coobservable sublanguages in decentralized supervisory control via communication," 2016, submitted, http://arxiv.org/abs/1512.03267.

[8] J. Komenda, T. Masopust, and J. H. van Schuppen, "On a distributed computation of supervisors in modular supervisory control," in *Int. Conference on Complex Systems Engineering (ICCSE)*, 2015, pp. 1–6.

[9] S. Lafortune and E. Chen, "The infimal closed controllable superlanguage and its application in supervisory control," *IEEE Trans. Automat. Control*, vol. 35, no. 4, pp. 398–405, 1990.

[10] K. Rudie and W. M. Wonham, "The infimal prefix-closed and observable superlanguange of a given language," *Systems Control Lett.*, vol. 15, no. 5, pp. 361–371, 1990.

[11] J. E. Hopcroft and J. D. Ullman, *Introduction to Automata Theory, Languages and Computation*. Addison-Wesley, 1979.

[12] M. Sipser, *Introduction to the theory of computation*, 2nd ed. Thompson Course Technology, 2006.

[13] S. Ginsburg, *Algebraic and Automata-theoretic Properties of Formal Languages*. Amsterdam: North-Holland, 1975.

[14] S. Yu, Q. Zhuang, and K. Salomaa, "The state complexities of some basic operations on regular languages," *Theoret. Comput. Sci.*, vol. 125, no. 2, pp. 315–328, 1994.

[15] G. Jirásková and T. Masopust, "On a structural property in the state complexity of projected regular languages," *Theoret. Comput. Sci.*, vol. 449, pp. 93–105, 2012.

[16] K. Wong, "On the complexity of projections of discrete-event systems," in *Proc. of WODES*, Cagliari, Italy, 1998, pp. 201–206.

[17] T. Cormen, C. Leiserson, R. Rivest, and C. Stein, *Introduction to Algorithms*, 3rd ed. MIT Press, 2009.

[18] T. Masopust, "Complexity of verifying nonblockingness in modular supervisory control," 2016, submitted. Preprint available online at http://math.cas.cz/masopust/pubs/Preprint/CompNonblockMSC.pdf.

[19] E. Bach and J. Shallit, *Algorithmic Number Theory, Volume I: Efficient Algorithms*. MIT Press, 1996.

[20] J.-C. Birget, "Intersection and union of regular languages and state complexity," *Inform. Process. Lett.*, vol. 43, pp. 185–190, 1992.

[21] N. J. A. Sloane, "The on-line encyclopedia of integer sequences," http://oeis.org. Sequence A002110. Accessed on October 18, 2016.