Online Appendix to: Querying Regular Graph Patterns

PABLO BARCELÓ, Universidad de Chile

LEONID LIBKIN, University of Edinburgh

JUAN L. REUTTER, University of Edinburgh and Pontificia Universidad Católica de Chile

This online appendix contains the following proofs and intermediate results.

- (1) Proof of Theorem 5.2
 - (2) Proof of Theorem 5.7
 - (3) Proof of Theorem 6.6
 - (4) Proof of Theorem 6.8
 - (5) Proof of Claim 7
-

PROOFS AND INTERMEDIATE RESULTS

Proof of Theorem 5.2

The only thing that is left is to prove Π_2^p -hardness for the class $\mathcal{P}^{\text{lv}}$. We reduce from the problem of $\forall\exists$ POSITIVE 1-3 3-SAT, which is known to be Π_2^p -hard [Björklund et al. 2007]. This problem is defined as follows: A set of clauses $\{C_1, \dots, C_p\}$ is given, each of which has exactly three distinct Boolean variables from the disjoint union of $\{x_1, \dots, x_m\}$ and $\{y_1, \dots, y_t\}$. No variable is negated. The problem asks whether, for each assignment for $\{x_1, \dots, x_m\}$, there exists an assignment for $\{y_1, \dots, y_t\}$ such that each clause C_i contains exactly one true variable.

Let $\phi := \forall x_1 \dots \forall x_m \exists y_1 \dots \exists y_t \{C_1, \dots, C_p\}$ be an instance of $\forall\exists$ positive 1-3 3-SAT. From ϕ , we construct in polynomial time an alphabet Σ , a pattern $\pi_\phi \in \mathcal{P}^{\text{lv}}$ and a CRPQ Q_ϕ , both over Σ , such that for each assignment for $\{x_1, \dots, x_m\}$, there exists an assignment for $\{y_1, \dots, y_t\}$ such that each clause C_i contains exactly one true variable if and only if $\text{CERTAIN}(Q_\phi, \pi_\phi) = \text{true}$.

We assume, without loss of generality, that ϕ contains clauses neither of the form $(x_j \vee x_k \vee x_\ell)$, for $1 \leq j < k < \ell \leq m$, nor of the form $(x_j \vee x_k \vee y_\ell)$, for $1 \leq j < k \leq m$ and $1 \leq \ell \leq t$. Indeed, it is clear that, if ϕ contains a clause of any of these forms, then there exists an assignment for $\{x_1, \dots, x_m\}$ such that, for no assignment for $\{y_1, \dots, y_t\}$, it is the case that each clause C_i contains exactly one true variable (this is the case, in particular, for the assignment that makes true each variable in $\{x_1, \dots, x_m\}$). This means that if ϕ contains clauses of any of these forms, then it does not belong to $\forall\exists$ positive 1-3 3-SAT. Further, it is easy to verify in polynomial time whether ϕ contains clauses of these forms.

The alphabet Σ over which π_ϕ is defined is:

$$\{\mathcal{C}_1, \dots, \mathcal{C}_p, (+), (-), (+-), (-+), (--), P, N, V, S, 0, 1\}.$$

The pattern π_ϕ uses label variables in the set $\{X_1, \dots, X_m\}$, and it is defined as follows.

First, π_ϕ contains node ids $n_\perp$ and $n_\top$ that represent, respectively, the Boolean values true and false. Intuitively, the fact that the node variable y_i ($1 \leq i \leq t$) of Q_ϕ , as

defined here, is assigned to node $n_{\perp}$ (respectively, $n_{\top}$) in a graph database $G \in \llbracket \pi_{\phi} \rrbracket$, represents a valuation of ϕ that assigns value false (respectively, true) to y_i .

In order to identify $n_{\perp}$ and $n_{\top}$ in π_{ϕ} , we mark the first node id with a self-loop labeled N (for *Negative* value) and the second one with a self-loop labeled P (for *Positive* value).

Second, for each clause C_i , $1 \leq i \leq p$, the pattern π_{C_i} contains a subpattern π_{C_i} that is defined by cases.

- (1) Assume first that $C_i = (y_j \vee y_{\ell} \vee x_k)$, where $1 \leq j < \ell \leq t$ and $1 \leq k \leq m$. Then, π_{C_i} consists of pairwise distinct node ids

$$n_{\perp}^i, n_{\top}^i, m_{\perp}^{i,1}, m_{\perp}^{i,2}, m_{\top}^{i,1}, m_{\top}^{i,2}, t_i.$$

Intuitively, the fact that the variable z_i^1 of Q_{ϕ} , as defined here, is mapped into node id $n_{\perp}^i$ (respectively, $n_{\top}^i$) in some graph database $G \in \llbracket \pi_{\phi} \rrbracket$, represents a valuation for ϕ that assigns value false (respectively, true) to the first variable of clause C_i (i.e., to y_j).

In the same way, the fact that the variable z_i^2 of Q_{ϕ} , as defined here, is mapped into node id $m_{\perp}^{i,1}$ (respectively, $m_{\top}^{i,2}$) in some graph database $G \in \llbracket \pi_{\phi} \rrbracket$, represents a valuation for ϕ that assigns value false (respectively, true) to both the first and the second variable of clause C_i (i.e., to y_j and y_{ℓ}). And, analogously, the fact that the variable z_i^2 of Q_{ϕ} , as defined here, is mapped into node id $m_{\perp}^{i,2}$ (respectively, $m_{\top}^{i,1}$) in some graph database $G \in \llbracket \pi_{\phi} \rrbracket$, represents a valuation for ϕ that assigns value true to the first variable, y_j , of clause C_i , and value false to the second one, y_{ℓ} (respectively, value false to the first variable of clause C_i and value true to the second one).

The following edges are the only edges that exist in between the node ids of π_{C_i} plus $\{n_{\perp}, n_{\top}\}$.

- Both $n_{\perp}^i, n_{\top}^i$ and t_i have a self-loop labeled C_i . This self-loop permits identifying these nodes as part of the pattern π_{C_i} .
- There are edges labeled V from $n_{\perp}^i$ into both $m_{\perp}^{i,1}$ and $m_{\top}^{i,1}$. These edges represent the fact that $n_{\perp}^i, m_{\perp}^{i,1}, m_{\top}^{i,1}$ are node ids that encode valuations for ϕ that assign value false to the first variable, y_j , of clause C_i .
- There are edges labeled V from $n_{\top}^i$ into both $m_{\perp}^{i,2}$ and $m_{\top}^{i,2}$. These edges represent the fact that $n_{\top}^i, m_{\perp}^{i,2}, m_{\top}^{i,2}$ are node ids that encode valuations for ϕ that assign value true to the first variable, y_j , of clause C_i .
- There are edges labeled S from $n_{\perp}^i, m_{\perp}^{i,1}$ and $m_{\perp}^{i,2}$ into $n_{\perp}$. These edges represent the fact that $n_{\perp}^i$ (respectively, $m_{\perp}^{i,1}$ and $m_{\perp}^{i,2}$) are node ids that encode valuations for ϕ that assign value false to the first (respectively, second) variable of clause C_i .
- There are edges labeled S from $n_{\top}^i, m_{\top}^{i,1}$ and $m_{\top}^{i,2}$ into $n_{\top}$. These edges represent the fact that $n_{\top}^i$ (respectively, $m_{\top}^{i,1}$ and $m_{\top}^{i,2}$) are node ids that encode valuations for ϕ that assign value true to the first (respectively, second) variable of clause C_i .
- The node id $m_{\perp}^{i,1}$ has a self-loop labeled $(--)$, that represents that $m_{\perp}^{i,1}$ encodes valuations for ϕ that assign the value false to the first two variables of C_i (i.e., to y_j and y_{ℓ}).

Fig. 7. Pattern π_{C_i} for a clause C_i of the form $(y_j \vee y_\ell \vee x_k)$.

- The node id $m_\perp^{i,2}$ has a self-loop labeled $(+-)$, that represents that $m_\perp^{i,2}$ encodes valuations for ϕ that assign value true to the first variable, y_j , of C_i and value false to the second one, y_ℓ .
- The node id $m_\top^{i,1}$ has a self-loop labeled $(-+)$, that represents that $m_\top^{i,1}$ encodes valuations for ϕ that assign value false to the first variable, y_j , of C_i and value true to the second one, y_ℓ .
- The node id $m_\top^{i,2}$ has a self-loop labeled $(++)$, that represents that $m_\top^{i,2}$ encodes valuations for ϕ that assign the value true to the first two variables of C_i (i.e., to y_j and y_ℓ).
- There are edges labeled X_k from each one of $m_\perp^{i,1}, m_\perp^{i,2}, m_\top^{i,1}$ and $m_\top^{i,2}$ into t_i . These edges represent the fact that the last variable of C_i is x_k .

Figure 7 shows how this pattern (together with all the edges that link this pattern to $n_\perp$ and $n_\top$) looks.

- (2) Assume now that $C_i = (y_j \vee y_\ell \vee y_k)$, where $1 \leq j < k < \ell \leq t$. Then, π_{C_i} is defined exactly as in the previous case, except that now the node id t_i is removed, together with all the edges that point to it, and the two new node ids $p_\perp^i, p_\top^i$ are added, together with the edges that link them to the rest of π_{C_i} and the nodes $n_\perp$ and $n_\top$ that we mention here.

In this case, the fact that the variable z_i^3 of Q_ϕ , as defined here, is mapped into node id $p_\perp^i$ (respectively, $p_\top^i$) in some graph database $G \in \llbracket \pi_\phi \rrbracket$, represents a valuation for ϕ that assigns value false (respectively, true) to the third variable of clause C_i (i.e., to y_k).

The pattern π_{C_i} contains the following edges linking the node ids in $\{p_\perp^i, p_\top^i\}$ to the rest of π_{C_i} and to the node ids $n_\perp$ and $n_\top$.

- The node ids $p_\perp^i$ and $p_\top^i$ and have a self-loop labeled C_i . This self-loop permits identifying these nodes as part of the pattern π_{C_i} .
- There is an edge labeled S from $p_\perp^i$ into $n_\perp$. This edge represents the fact that $p_\perp^i$ is the node id that encodes valuations for ϕ that assign value false to the third variable, y_k , of clause C_i .

where the regular language $\mathcal{R}$ is defined as follows:

$$\mathcal{R} := ((--)\cdot P \cup (-+)\cdot N \cup (+-)\cdot N) \cdot \mathcal{C}_i.$$

Clearly, Q_ϕ can be constructed in polynomial time from ϕ .

We prove next that ϕ belongs to $\forall\exists$ POSITIVE 1-3 3SAT if and only if $\text{CERTAIN}(Q_\phi, \pi_\phi) = \text{true}$. Assume first that $\phi = \forall x_1 \dots \forall x_m \exists y_1 \dots \exists y_t \{C_1, \dots, C_p\}$ belongs to $\forall\exists$ POSITIVE 1-3 3SAT. Consider an arbitrary graph database $G \in \llbracket \pi_\phi \rrbracket$. Then, there exists a homomorphism $h : \pi_\phi \rightarrow G$. Since π_ϕ does not contain node variables, we assume, without loss of generality, that h is a mapping from the label variables of π_ϕ , that is, $\{X_1, \dots, X_m\}$, into Σ . We prove next that Q_ϕ holds in G , which shows, in turn, that $\text{CERTAIN}(Q_\phi, \pi_\phi) = \text{true}$ (since G was arbitrarily chosen).

Let $\mathcal{X}$ be the set of label variables in $\{X_1, \dots, X_m\}$ such that $h(X) = 1 \Leftrightarrow X \in \mathcal{X}$. (Notice that $\mathcal{X}$ can be empty). Consider the assignment κ for the propositional variables $\{x_1, \dots, x_m\}$ of ϕ into Boolean values true and false, such that κ assigns value true to variable x_i if and only if $X_i \in \mathcal{X}$ ($1 \leq i \leq m$). Since ϕ belongs to $\forall\exists$ POSITIVE 1-3 3SAT, there is an assignment κ' for the propositional variables $\{x_1, \dots, x_m, y_1, \dots, y_t\}$ of ϕ into Boolean values true and false, such that κ' coincides with κ on $\{x_1, \dots, x_m\}$ and κ' assigns value true to exactly one propositional variable in each clause C_i , $1 \leq i \leq p$.

Recall that the set of node variables of Q_ϕ is $\{y_1, \dots, y_t, z_1^1, z_1^2, z_1^3, \dots, z_p^1, z_p^2, z_p^3\}$. We define σ to be any mapping from $\{y_1, \dots, y_t, z_1^1, z_1^2, z_1^3, \dots, z_p^1, z_p^2, z_p^3\}$ into the node ids of G that satisfies the following.

- For each $1 \leq i \leq t$, $\sigma(y_i) = n_\perp$ if $\kappa'(y_i)$ is false, and $\sigma(y_i) = n_\top$ otherwise.
- If C_i is of the form $(y_j \vee y_\ell \vee x_k)$, for $1 \leq j < \ell \leq t$ and $1 \leq k \leq m$, then:
 - (1) $\sigma(z_i^1) = n_\perp^i$ if $\sigma(y_j) = n_\perp$, and $\sigma(z_i^1) = n_\top^i$ otherwise;
 - (2) $\sigma(z_i^2) = m_\perp^{i,1}$ if $\sigma(y_j) = \sigma(y_\ell) = n_\perp$, $\sigma(z_i^2) = m_\perp^{i,2}$ if $\sigma(y_j) = n_\top$ and $\sigma(y_\ell) = n_\perp$,
 $\sigma(z_i^2) = m_\top^{i,1}$ if $\sigma(y_j) = n_\perp$ and $\sigma(y_\ell) = n_\top$, and $\sigma(z_i^2) = m_\top^{i,2}$ otherwise;
 - (3) $\sigma(z_i^3) = t_i$.
- If C_i is of the form $(y_j \vee y_\ell \vee y_k)$, for $1 \leq j < \ell < k \leq t$, then:
 - (1) $\sigma(z_i^1) = n_\perp^i$ if $\sigma(y_j) = n_\perp$, and $\sigma(z_i^1) = n_\top^i$ otherwise;
 - (2) $\sigma(z_i^2) = m_\perp^{i,1}$ if $\sigma(y_j) = \sigma(y_\ell) = n_\perp$, $\sigma(z_i^2) = m_\perp^{i,2}$ if $\sigma(y_j) = n_\top$ and $\sigma(y_\ell) = n_\perp$,
 $\sigma(z_i^2) = m_\top^{i,1}$ if $\sigma(y_j) = n_\perp$ and $\sigma(y_\ell) = n_\top$, and $\sigma(z_i^2) = m_\top^{i,2}$ otherwise;
 - (3) $\sigma(z_i^3) = p_\perp^i$ if $\sigma(y_k) = n_\perp$, and $\sigma(z_i^3) = p_\top^i$ otherwise.

It is easy to see that assignment σ is well defined. We prove next that $(G, \sigma) \models Q_\phi$, and hence that Q_ϕ holds in G .

Recall that Q_ϕ consists of conjuncts $(y_i, (P \cup N), y_i)$, for each $1 \leq i \leq t$, and conjunctions θ_{C_i} , for each $1 \leq i \leq p$. Clearly, $(G, \sigma) \models (y_i, (P \cup N), y_i)$, for each $1 \leq i \leq t$. This is because $\sigma(y_i)$ is either $n_\perp$ and $n_\top$, which have self-loops labeled N and P , respectively. We prove next that $(G, \sigma) \models \theta_{C_i}$, for each $1 \leq i \leq p$.

Assume first that C_i is of the form $(y_j \vee y_\ell \vee x_k)$, for $1 \leq j < \ell \leq t$ and $1 \leq k \leq m$. Recall that in this case θ_{C_i} is defined as

$$(z_i^1, S, y_j), (z_i^2, S, y_\ell), (z_i^1, C_i \cdot V, z_i^2), (z_i^2, \mathcal{R}, z_i^3),$$

where the regular language $\mathcal{R}$ is defined as follows:

$$\mathcal{R} := \left(\bigcup_{a \in \Sigma \setminus \{0,1\}} a \right) \cdot \mathcal{C}_i \cup ((--)\cdot 1 \cup (-+)\cdot 0 \cup (+-)\cdot 0) \cdot \mathcal{C}_i.$$

It is easy to see that, by definition, $(G, \sigma) \models (z_i^1, S, y_j), (z_i^2, S, y_\ell)$. This is because, in every possible case, $\sigma(z_i^1)$ (respectively, $\sigma(z_i^2)$) corresponds to a node id that can access $\sigma(y_j)$ (respectively, $\sigma(y_\ell)$) through an S -labeled edge. Further, clearly $(G, \sigma) \models (z_i^1, C_i \cdot V, z_i^2)$. This is because $\sigma(z_i^1)$ is either $n_\perp^i$ or $n_\top^i$, which have C_i -labeled self-loops, and in every possible case $\sigma(z_i^1)$ corresponds to a node id that can access $\sigma(z_i^2)$ through a V -labeled edge. We prove next by cases that $(G, \sigma) \models (z_i^2, \mathcal{R}, z_i^3)$.

- (1) Assume first that $h(X_i) \notin \{0, 1\}$. Then, clearly $(G, \sigma) \models (z_i^2, (\bigcup_{a \in \Sigma \setminus \{0,1\}} a) \cdot C_i, z_i^3)$. This is because $\sigma(z_i^3) = t_i$, t_i has a self-loop labeled C_i , and, in every possible case, there is an edge from $\sigma(z_i^2)$ into $\sigma(z_i^3) = t_i$ labeled with $h(X_i) \in \bigcup_{a \in \Sigma \setminus \{0,1\}} a$.
- (2) Assume second that $h(X_i) = 1$. Then, $\kappa'(x_i)$ is true, and hence since κ' makes true exactly one propositional variable in C_i , it must be the case that $\kappa'(y_j) = \kappa'(y_\ell) = \text{false}$, which implies that $\sigma(y_j) = \sigma(y_\ell) = n_\perp$. This implies that $\sigma(z_i^2) = m_\perp^{i,1}$, and hence that $(G, \sigma) \models (z_i^2, (--) \cdot 1 \cdot C_i, z_i^3)$. This is because $\sigma(z_i^2) = m_\perp^{i,1}$ has a self-loop labeled $(--)$, there is an edge from $\sigma(z_i^2)$ into $\sigma(z_i^3) = t_i$ labeled with $h(X_i) = 1$, and t_i has a self-loop labeled C_i .
- (3) The case when $h(X_i) = 0$ is similar and left to the reader.

In any case, we conclude that $(G, \sigma) \models (z_i^2, \mathcal{R}, z_i^3)$.

Assume then that C_i is of the form $(y_j \vee y_\ell \vee y_k)$, for $1 \leq j < \ell < k \leq t$. Recall that, in this case, θ_{C_i} is defined as

$$(z_i^1, S, y_j), (z_i^2, S, y_\ell), (z_i^3, S, y_k), (z_i^1, C_i \cdot V, z_i^2), (z_i^2, \mathcal{R}, z_i^3),$$

where the regular language $\mathcal{R}$ is defined as follows:

$$\mathcal{R} := ((--)) \cdot P \cup (++) \cdot N \cup (+-) \cdot N \cdot C_i.$$

As in the previous case, it is easy to see that, by definition, $(G, \sigma) \models (z_i^1, S, y_j), (z_i^2, S, y_\ell), (z_i^3, S, y_k)$. This is because, in every possible case, $\sigma(z_i^1)$ (respectively, $\sigma(z_i^2)$ and $\sigma(z_i^3)$) corresponds to a node id that can access $\sigma(y_j)$ (respectively, $\sigma(y_\ell)$ and $\sigma(y_k)$) through an S -labeled edge. Further, clearly $(G, \sigma) \models (z_i^1, C_i \cdot V, z_i^2)$. This is because $\sigma(z_i^1)$ is either $n_\perp^i$ or $n_\top^i$, which have C_i -labeled self-loops, and in every possible case $\sigma(z_i^1)$ corresponds to a node id that can access $\sigma(z_i^2)$ through a V -labeled edge. We prove next, by cases, that $(G, \sigma) \models (z_i^2, \mathcal{R}, z_i^3)$.

- (1) Assume first that $\kappa'(x_k) = \text{true}$. This implies that $\sigma(x_k) = p_\top^i$, and hence that $(G, \sigma) \models (z_i^2, (--) \cdot P \cdot C_i, z_i^3)$. Indeed, since κ' makes true exactly one variable in C_i it must be the case that $\kappa'(y_j) = \kappa'(y_\ell) = \text{false}$, and hence $\sigma(y_j) = \sigma(y_\ell) = n_\perp$. This implies that $\sigma(z_i^2) = m_\perp^{i,1}$, and hence that $(G, \sigma) \models (z_i^2, (--) \cdot P \cdot C_i, z_i^3)$. This is because $\sigma(z_i^2) = m_\perp^{i,1}$ has a self-loop labeled $(--)$, there is an edge labeled P from $\sigma(z_i^2)$ into $\sigma(z_i^3) = p_\top^i$, and $p_\top^i$ has a self-loop labeled C_i .
- (2) The case when $\kappa'(x_k) = \text{false}$ is similar and left to the reader.

In any case, we conclude that $(G, \sigma) \models (z_i^2, \mathcal{R}, z_i^3)$.

Assume now, on the other hand, that $\text{CERTAIN}(Q_\phi, \pi_\phi) = \text{true}$. Take an arbitrary valuation κ from the propositional variables $\{x_1, \dots, x_k\}$ into Boolean values true and

false. We prove next that there is an assignment κ' for the propositional variables $\{x_1, \dots, x_m, y_1, \dots, y_t\}$ of ϕ into Boolean values true and false, such that κ' coincides with κ on $\{x_1, \dots, x_m\}$ and κ' assigns value true to exactly one propositional variable in each clause C_i , $1 \leq i \leq p$. This is sufficient to prove that ϕ belongs to $\forall\exists$ POSITIVE 1-3 3SAT since κ is arbitrarily chosen.

Consider the graph database G over Σ that is obtained from π_ϕ by replacing, for each $1 \leq i \leq p$, the label variable X_i with 1, if $\kappa(X_i) = \text{true}$, and with 0, otherwise. Clearly, $G \in \llbracket \pi_\phi \rrbracket$ and hence Q_ϕ holds in G . Assume then that σ is an assignment for the node variables $\{y_1, \dots, y_t, z_1^1, z_1^2, z_1^3, \dots, z_p^1, z_p^2, z_p^3\}$ of Q_ϕ such that $(G, \sigma) \models Q_\phi$. Let us define κ' to be the following assignment for the propositional variables $\{x_1, \dots, x_m, y_1, \dots, y_t\}$ of ϕ : $\kappa'(x_i) = \kappa(x_i)$, for each $1 \leq i \leq m$, and for each $1 \leq j \leq t$, it is the case $\kappa'(y_j) = \text{true}$ if $\sigma(y_j) = n_\top$, and $\kappa'(y_j) = \text{false}$, otherwise. We prove next that κ' makes true exactly one propositional variable in each clause C_i , for $1 \leq i \leq p$.

Assume first that C_i is of the form $(y_j \vee y_\ell \vee x_k)$, for $1 \leq j < \ell \leq t$ and $1 \leq k \leq m$. Suppose initially that $\kappa(x_k) = \text{true}$. Then, every edge that leads from $\sigma(z_i^2)$ into $\sigma(z_i^3)$ in G is labeled with 1 (since it was labeled with X_k in π_ϕ). Since $(G, \sigma) \models Q_\phi$, it is the case that $(G, \sigma) \models (z_i^2, \mathcal{R}, z_i^3)$, where $\mathcal{R} = (\bigcup_{a \in \Sigma \setminus \{0,1\}} a) \cdot C_i \cup ((--)\cdot 1 \cup (-+)\cdot 0 \cup (+-)\cdot 0) \cdot C_i$. But the only possibility for this to happen in this case is that $(G, \sigma) \models (z_i^2, (--)\cdot 1 \cdot C_i, z_i^3)$. Notice that this immediately implies that $\sigma(z_i^2) = m_{\perp}^{i,1}$, and hence that $\sigma(y_j) = \sigma(y_\ell) = n_\perp$. We conclude that $\kappa'(y_j) = \kappa'(y_\ell) = \text{false}$, and, therefore, κ' only makes true one propositional variable in C_i . The case when $\kappa(x_k) = \text{false}$ can be handled analogously.

The case when C_i is of the form $(y_j \vee y_\ell \vee x_k)$, for $1 \leq j < \ell \leq t$ and $1 \leq k \leq m$ is analogous and left to the reader.

This finishes the proof of the theorem. It is interesting to notice that although the proof just presented uses a nonfixed alphabet (in particular, dependent on the number of clauses in the propositional formula ϕ), one can quite easily come out with a refinement of this reduction that uses a fixed alphabet. We decided to show here the simpler reduction, with a non-fixed alphabet, for the sake of presentation and readability.

Proof of Theorem 5.7

We only have to prove the second part. That is, we prove that there exists a Boolean RPQ Q of the form $\text{Ans}() \leftarrow (x, w, y)$, where w is a single word, such that $\text{DATA COMPLEXITY}(Q)$ is CONP-hard even over input patterns in $\mathcal{P}_{\text{Codd}}^{\text{lv}}$ whose underlying graph is a DAG. We work with the word $w = 1011011101111$.

The proof is by a reduction from 3SAT to the complement of $\text{DATA COMPLEXITY}(Q)$. For the sake of readability, we first construct in polynomial time from each propositional formula ϕ in 3CNF a pattern $\pi \in \mathcal{P}_{\text{Codd}}^{\text{lv, re}}$ over alphabet $\{0, 1\}$, and show that ϕ is satisfiable if and only if $\text{CERTAIN}(Q, \pi) = \text{false}$. Afterwards, we show how to construct in polynomial time from π a pattern $\pi' \in \mathcal{P}_{\text{Codd}}^{\text{lv}}$ over alphabet $\{0, 1\}$, such that $\text{CERTAIN}(Q, \pi) = \text{false}$ iff $\text{CERTAIN}(Q, \pi') = \text{false}$.

Let $\phi = \bigwedge_{1 \leq j \leq m} (c_j^1 \vee c_j^2 \vee c_j^3)$ be a propositional formula in 3CNF that uses propositional variables from the set $\{x_1, \dots, x_k\}$; that is, each c_j^i , for $1 \leq j \leq m$ and $1 \leq i \leq 3$, is either a variable x_ℓ , $1 \leq \ell \leq k$, or its negation. We associate with each propositional variable x_ℓ , $1 \leq \ell \leq k$, a fresh label variable X_ℓ . Also, with each clause $(c_j^1 \vee c_j^2 \vee c_j^3)$, $1 \leq j \leq m$ we associate three fresh label variables C_j^1, C_j^2 and C_j^3 .

Fig. 9. Pattern π for the case when $\phi = (x_1 \vee x_2 \vee \neg x_3) \wedge (x_2 \vee x_3 \vee \neg x_4)$.

The pattern π over alphabet $\{0, 1\}$ contains the following edges.

- For each $1 \leq j \leq m$, π contains the edges $(n_j^1, 1, n_j^2)$, (n_j^2, C_j^1, n_j^3) , $(n_j^3, 11, n_j^4)$, (n_j^4, C_j^2, n_j^5) , $(n_j^5, 111, n_j^6)$, (n_j^6, C_j^3, n_j^7) , $(n_j^7, 1111, n_j^8)$, where $n_j^1, \dots, n_j^8$ are fresh node ids.
- For each $1 \leq \ell \leq k$, π contains the edges $(p_\ell^1, X_\ell, p_\ell^2)$ and $(p_\ell^2, 1111, p_\ell^3)$, where $p_\ell^1, p_\ell^2, p_\ell^3$ are fresh node ids.
- If $c_j^i = x_\ell$, for $1 \leq j \leq m$, $1 \leq i \leq 3$ and $1 \leq \ell \leq k$, the pattern π contains the edges $(n_j^{(2i+1)}, 1101110, p_\ell^1)$ and $(n_j^{(2i+1)}, 0110111, p_\ell^1)$.
- If $c_j^i = \neg x_\ell$, for $1 \leq j \leq m$, $1 \leq i \leq 3$ and $1 \leq \ell \leq k$, the pattern π contains the edges $(n_j^{(2i+1)}, 110111, p_\ell^1)$ and $(n_j^{(2i+1)}, 01101110, p_\ell^1)$.

Clearly, π belongs to $\mathcal{P}_{\text{Codd}}^{\text{lv, re}}$ and can be constructed in polynomial time from ϕ . Furthermore, it is easy to see that the underlying graph of π is a DAG. Figure 9 shows how this pattern looks for the case when $\phi = (x_1 \vee x_2 \vee \neg x_3) \wedge (x_2 \vee x_3 \vee \neg x_4)$. We have skipped the names of node ids since they are all different and can be easily inferred from the context.

We prove next that ϕ is satisfiable if and only if $\text{CERTAIN}(Q, \pi) = \text{false}$.

($\Rightarrow$) Assume first that ϕ is satisfiable via assignment $\gamma : \{x_1, \dots, x_k\} \rightarrow \{0, 1\}$. In order to prove that $\text{CERTAIN}(Q, \pi) = \text{false}$, we show the existence of a graph database G

over alphabet $\{0, 1\}$ such that $G \not\models Q$ but $G \models \pi$. To define G , we use again the notion of *canonical* graph database for π that we developed in the proof of Proposition 5.6.

Let σ be a canonical assignment for π , and define the following mapping ν from the label variables of π to $\{0, 1\}$.

- For each $1 \leq \ell \leq k$, define $\nu(X_\ell) = 1$ if and only if $\gamma(x_\ell) = 1$.
- For each $1 \leq j \leq m$ and $1 \leq i \leq 3$ define $\nu(C_j^i) = 1$ if and only if $\gamma(x_\ell) = 1$, if the i th literal of the j th clause of ϕ corresponds to x_ℓ for some $1 \leq \ell \leq k$, or $\nu(C_j^i) = 1$ if and only if $\gamma(x_\ell) = 0$, if the i th literal of the j th clause of ϕ corresponds to $\neg x_\ell$ for some $1 \leq \ell \leq k$.

We then define G as the σ -canonical graph for π that is witnessed by ν (it is trivial to show that such graph always exists). Given that G is canonical for π , we obtain that $G \models \pi$.

Next we prove that $G \not\models Q$. We first claim that there is no path in G labeled w (recall that $w = 1011011101111$) that contains some vertex p_ℓ^1 , for $1 \leq \ell \leq k$. Assume for the sake of contradiction that there exists a path ρ such that $\lambda(\rho) = 1011011101111$ and ρ contains the node id p_ℓ^1 , for some $1 \leq \ell \leq k$. Notice that $|\rho| = 14$, since $|w| = 13$. It is not hard to see that G is a DAG, and, thus, the fourteen node ids in ρ need to be distinct. We consider several cases.

- The path does not use node ids in the set $\{n_j^1, \dots, n_j^8 \mid 1 \leq j \leq m\}$. But from the properties of G , we know that this case is simply not possible. Indeed, the path in G from any node in the set $\{n_j^1, \dots, n_j^8 \mid 1 \leq j \leq m\}$ to node p_ℓ^1 contains at most seven nodes from N (since each of these paths can only be labeled with a word of at most seven letters). Further, any path starting in p_ℓ^1 can have at most six nodes (going to p_ℓ^2 and then to p_ℓ^3). Thus, a path in G that uses no node in the set $\{n_j^1, \dots, n_j^8 \mid 1 \leq j \leq m\}$ but goes through p_ℓ^1 uses at most thirteen distinct node ids.
- The path does not contains a vertex $n_j^{(2i+1)}$, for some $1 \leq j \leq m$ and $1 \leq i \leq 3$, such that c_j^i is either x_ℓ or $\neg x_\ell$. This case is also not possible, given that the only edges that lead into node p_ℓ^1 are from nodes of the form $n_j^{(2i+1)}$, for some $1 \leq j \leq m$ and $1 \leq i \leq 3$, such that c_j^i is either x_ℓ or $\neg x_\ell$.
- The path contains a vertex $n_j^{(2i+1)}$, for some $1 \leq j \leq m$ and $1 \leq i \leq 3$, such that $c_j^i = x_\ell$. Then, from the definition of ν , we can check that $\nu(C_j^i) = \nu(X_\ell)$. From the properties of G , there are only two paths from $n_j^{(2i+1)}$ to p_ℓ^1 : The first one is labeled with 1101110, and the second one is labeled with 0110111. Thus, either $\nu(C_j^i) \cdot 1101110 \cdot \nu(X_\ell)$ is a subword of $w = 1011011101111$, or $\nu(C_j^i) \cdot 0110111 \cdot \nu(X_\ell)$ is a subword of $w = 1011011101111$. But it is not hard to see that this cannot be the case since $\nu(C_j^i) = \nu(X_\ell)$.
- The path contains a vertex $n_j^{(2i+1)}$, for some $1 \leq j \leq m$ and $1 \leq i \leq 3$, such that $c_j^i = \neg x_\ell$. By using the same arguments as the previous case, one can show that in this case it must hold that $\nu(C_j^i) \cdot 110111 \cdot \nu(X_\ell)$ is a subword of $w = 1011011101111$ or $\nu(C_j^i) \cdot 01101110 \cdot \nu(X_\ell)$ is a subword of $w = 1011011101111$. It is not hard to see that this cannot be the case, since ν is defined in such a way that $\nu(C_j^i) = 1$ if and only if $\nu(X_\ell) = 0$.

We have proved that ρ does not contain any vertex of the form p_ℓ^1 , for $1 \leq \ell \leq k$. We prove next that if we assume, for the sake of contradiction, that there is a path ρ in G such that $\lambda(\rho) = 1011011101111$, then it must be the case that ρ contains all nodes $n_j^1, \dots, n_j^8$, for some $1 \leq j \leq m$. First, by a simple counting argument, it is easy to see that ρ cannot end in a node of the form n_j^s , for $1 \leq s \leq 7$ (because ρ uses at least fourteen distinct node ids). Thus, since ρ cannot contain any node of the form p_ℓ^1 , $1 \leq \ell \leq k$, the only remaining option is that ρ ends in a node from G that is in a path that connects a node of the form $n_j^{(2i+1)}$, $1 \leq j \leq m$ and $1 \leq i \leq 3$, to a node of the form p_ℓ^1 , for $1 \leq \ell \leq k$. But since w ends with 4 consecutive 1's, the only possibility is that ρ ends in the first two nodes of a path ρ' that connects a node of the form $n_j^{(2i+1)}$, $1 \leq j \leq m$ and $1 \leq i \leq 3$, to a node of the form p_ℓ^1 , and such that ρ' is labeled either by 1101110 or 110111. But then this means that there is a path labeled 01 that leads in G to n_j^{2i} , which clearly cannot happen.

Thus, ρ is a path that goes from node id n_j^1 to node id n_j^8 , for some $1 \leq j \leq m$. From the construction of G , this implies that $v(C_j^1) = v(C_j^2) = v(C_j^3) = 0$, which in turn implies that ϕ is not satisfied by valuation γ . This is the desired contradiction.

($\Leftarrow$) Assume, on the other hand, that $\text{CERTAIN}(Q, \pi) = \text{false}$. Then, there is a graph G such that $G \models \pi$ but $G \not\models Q$. Since $G \models \pi$, there is a homomorphism $h = (h_1, h_2)$ from π to G , such that h_1 is the identity map on the node ids of π , and h_2 is a mapping from the label variables used in π to the alphabet $\{0, 1\}$. From h_2 , we construct a valuation γ for the propositional variables of ϕ as follows: $\gamma(x_i) = 1$ if and only if $h_2(X_i) = 1$, for each $1 \leq i \leq k$. We show next that ϕ is satisfied by assignment γ . In order to prove this, we use the following claim (for simplicity, we say that $\gamma(c_j^i)$ is the valuation of the i th literal of the j th clause of ϕ).

CLAIM 9. *For each $1 \leq j \leq m$ and $1 \leq i \leq 3$, it is the case that $\gamma(c_j^i) = 1$ if and only if $h_2(C_j^i) = 1$.*

Before proving the claim, we show how it follows from it that valuation γ satisfies the propositional formula ϕ . Indeed, assume on the contrary that γ does not satisfy ϕ , that is, for some $1 \leq j \leq m$, it is the case that $\gamma(c_j^1) = \gamma(c_j^2) = \gamma(c_j^3) = 0$. Then, from Claim 9, it follows that $h_2(C_j^1) = h_2(C_j^2) = h_2(C_j^3) = 0$, and, thus, it is easy to see that there is a path ρ from n_j^1 to n_j^8 in G that satisfies that $\lambda(\rho) = w = 1011011101111$. But this contradicts the fact that $G \not\models Q$.

PROOF OF CLAIM 9. Assume, for the sake of contradiction, that for some $1 \leq j \leq m$ and $1 \leq i \leq 3$, it is the case that $\gamma(c_j^i) \neq h_2(C_j^i)$. We consider four cases depending on h_2 and the choices of i and j .

- It holds that $h_2(C_j^i) = 0$ and $c_j^i = x_\ell$ for some $1 \leq \ell \leq k$. Then, $\gamma(c_j^i) = 1$, $\gamma(x_\ell) = 1$ and, from the definition of γ , $h_2(X_\ell) = 1$. Further, since $G \models \pi$, we know that the following hold: (1) There is a path in G that goes from $n_j^{(2i-1)}$ to n_j^{2i} that is labeled with a word with i consecutive 1's; (2) There is a path in G that goes from n_j^{2i} to $n_j^{(2i+1)}$ that is labeled with $h_2(C_j^i) = 0$; (3) There is a path in G from $n_j^{(2i+1)}$ to p_ℓ^1 that is labeled with 1101110; (4) There is a path in G from p_ℓ^1 to p_ℓ^2 that is

- labeled $h_2(X_\ell) = 1$; (5) There is a path in G from p_ℓ^2 to p_ℓ^3 that is labeled with 1111. Combining all these paths, it is easy to see that there is a path in G that is labeled with $w = 1011011101111$. This is a contradiction, as we have assumed that $G \not\models Q$.
- It holds that $h_2(C_j^i) = 1$, and $c_j^i = x_\ell$ for some $1 \leq \ell \leq k$. This case is analogous to the previous one, but this time, we use the fact that there is a path in G from $n_j^{(2i+1)}$ to p_ℓ^1 that is labeled with 0110111.
 - We have that $h_2(C_j^i) = 0$ and $c_j^i = \neg x_\ell$ for some $1 \leq \ell \leq k$. Then, $\gamma(c_j^i) = 1$, $\gamma(x_\ell) = 0$, and from the definition of γ , $h_2(X_\ell) = 0$. Further, since $G \models \pi$, we know that the following hold: (1) There is a path in G that goes from $n_j^{(2i-1)}$ to n_j^{2i} that is labeled with a word with i consecutive 1's; (2) There is a path in G that goes from n_j^{2i} to $n_j^{(2i+1)}$ that is labeled with $h_2(C_j^i) = 0$; (3) There is a path in G from $n_j^{(2i+1)}$ to p_ℓ^1 that is labeled with 110111; (4) There is a path in G from p_ℓ^1 to p_ℓ^2 that is labeled $h_2(X_\ell) = 0$; (5) There is a path in G from p_ℓ^2 to p_ℓ^3 that is labeled with 1111. Combining all these paths, it is easy to see that there is a path in G that is labeled with $w = 1011011101111$. This is a contradiction, as we have assumed that $G \not\models Q$.
 - It holds that $h_2(C_j^i) = 1$, and $c_j^i = \neg x_\ell$ for some $1 \leq \ell \leq k$. This case is analogous to the previous one, but this time we use the fact that there is a path in G from $n_j^{(2i+1)}$ to p_ℓ^1 that is labeled with 01101110. □

This finishes the proof that ϕ is satisfiable if and only if $\text{CERTAIN}(Q, \pi) = \text{false}$.

To complete the proof, we need to show that there is a polynomial-time procedure that, from π , constructs a pattern π' in $\mathcal{P}_{\text{Codd}}^{\text{lv}}$ such that $\text{CERTAIN}(Q, \pi) = \text{false}$ iff $\text{CERTAIN}(Q, \pi') = \text{false}$. But this is quite straightforward, since each regular language that is used to label an edge in π consists of a single word, and, thus, we can obtain an equivalent pattern π' by replacing each edge labeled with the word w with a path ρ of fresh node ids such that $\lambda(\rho) = w$. This finishes the proof of the theorem.

Proof of Theorem 6.6

Only the first part is missing. Let $Q = \text{Ans}(x, y, \rho) \leftarrow (x, \rho : L, y)$ be an RPQ with a path output. Also, let $\pi = (N, E)$ be a graph pattern, $\mathcal{W}$ the set of label variables mentioned in π , and n_1, n_2 two of its nodes from $\mathcal{V}$. We prove that $(n_1, n_2) \in \text{CERTAIN}(Q, \pi)$ if and only if L is weakly accepted by $\mathcal{A}_\pi(n_1, n_2)$.

($\Rightarrow$): Assume that $(n_1, n_2) \in \text{CERTAIN}(Q, \pi)$. We prove that L is weakly accepted by $\mathcal{A}_\pi(n_1, n_2)$. Let $v = (\eta, \theta)$ be an arbitrary valuation for $\mathcal{A}_\pi(n_1, n_2)$; that is, η is a mapping from $\mathcal{W}$ into Σ and $\theta : (N \times \text{REG}(\Sigma \cup \mathcal{W}) \times N) \rightarrow (N \times \Sigma^* \times N)$ assigns to each edge $(p, r, q) \in E$ a transition (p, w, q) , where w is a word that belongs to $\eta(r)$. Next, we show that $L \cap L(v(\mathcal{A}_\pi(n_1, n_2))) \neq \emptyset$.

Let σ be the assignment from the nodes of π into $\mathcal{V}$ that is the identity on node ids and maps each node variable x into a different node id n_x . Then, we define a graph database G as the unique (up to isomorphism) σ -canonical graph database for π that satisfies the following: For every edge $e = (p, r, q)$ of π , the path ρ that is associated with e in G is such that $\lambda(\rho) = w$, where $\theta(e) = (p, w, q)$. Notice that G is, indeed, a σ -canonical assignment via η . This is because for each edge $e = (p, r, q)$ in E it is the case that if $\theta(e) = (p, w, q)$ then $w \in \eta(r)$.

It is immediately clear that $G \in \llbracket \pi \rrbracket$ (since G is σ -canonical for π), and that $\sigma(n_1) = n_1$ and $\sigma(n_2) = n_2$. Furthermore, since $(n_1, n_2) \in \text{CERTAIN}(Q, \pi)$, it is the case that

$(n_1, n_2) \in Q(G)$. Thus, there is a path ρ in G from n_1 to n_2 such that $\lambda(\rho) \in L$. It is now easy to show that there is a run of $v(\mathcal{A}_\pi(n_1, n_2))$ that accepts a word in L (namely, the word $\lambda(\rho)$). This is because the transitions of $v(\mathcal{A}_\pi(n_1, n_2))$ are precisely the paths of G that are associated with the edges of π ; that is, if (p, u, q) is a transition in $v(\mathcal{A}_\pi(n_1, n_2))$ then there is a path in G from p to q labeled u .

Thus, $L \cap L(v(\mathcal{A}_\pi(n_1, n_2))) \neq \emptyset$. Since v is an arbitrary valuation, we conclude that L is weakly accepted by $\mathcal{A}_\pi(n_1, n_2)$.

($\Leftarrow$): Assume that L is weakly accepted by $\mathcal{A}_\pi(n_1, n_2)$. We prove that $(n_1, n_2) \in \text{CERTAIN}(Q, \pi)$. From Claim 2, we only need to show that for every canonical graph database G for π , it is the case that $(n_1, n_2) \in Q(G)$. This is what we do next.

Let G be an arbitrary graph database $G \in \llbracket \pi \rrbracket$, and assume that G is σ -canonical for π via assignment $\eta : \mathcal{W} \rightarrow \Sigma$. (Recall that σ is an assignment from the nodes of π into $\mathcal{V}$ that is the identity on node ids and maps each node variable x into a different node id n_x .) Clearly, both node ids n_1 and n_2 belong to G . For each edge e in π , let us denote by ρ_e the path of G that is associated with e .

Let us define now a mapping $\theta : (N \times \text{REG}(\Sigma \cup \mathcal{W}) \times N) \rightarrow (N \times \Sigma^* \times N)$ that assigns to each edge $e = (p, r, q) \in E$ a transition (p, w, q) , where w is the word $\lambda(\rho_e)$. This is clearly well defined, since, by definition, $\lambda(\rho_e)$ satisfies $\eta(r)$.

Since L is weakly accepted by $\mathcal{A}_\pi(n_1, n_2)$, it is the case that $L \cap L(v(\mathcal{A}_\pi(n_1, n_2))) \neq \emptyset$. Thus, there is a word $w \in L$ that is accepted by $v(\mathcal{A}_\pi(n_1, n_2))$. It is not hard to prove then that there is a path in G from n_1 to n_2 that is labeled w . This is because the transitions of $v(\mathcal{A}_\pi(n_1, n_2))$ are precisely the paths of G that are associated with the edges of π ; that is, if (p, u, q) is a transition in $v(\mathcal{A}_\pi(n_1, n_2))$ then there is a path in G from p to q labeled u .

This implies that $(n_1, n_2) \in Q(G)$. Since G is an arbitrary graph database that is canonical for π , we conclude that $(n_1, n_2) \in \text{CERTAIN}(Q, \pi)$. $\square$

Proof of Theorem 6.8

We provide the complete proof of the fact that $L_s(\mathcal{A}_1) \cap \bigcap_{A \in \mathbb{A}_2, \mathbb{A}_3, \mathbb{A}_4} L(A)$ is not empty if and only if $\mathcal{M}$ accepts on input $\bar{a}$.

($\Rightarrow$) Assume that there is a word w that belongs to $L_s(\mathcal{A}_1) \cap \bigcap_{A \in \mathbb{A}_2, \mathbb{A}_3, \mathbb{A}_4} L(A)$. We show that $\mathcal{M}$ accepts in input $\bar{a}$ by constructing an accepting run for $\mathcal{M}$, as follows.

Since w belongs to $\bigcap_{A \in \mathbb{A}_2, \mathbb{A}_3, \mathbb{A}_4} L(A)$ it must be of the form:

$$(000(0 \mid 1)^n)^* \left(001(0 \mid 1)^n 001(0 \mid 1)^n (010 \mid 100 \mid 101)[0] \cdots (010 \mid 100 \mid 101)[2^n - 1] \right)^* 111(0 \mid 1)^n.$$

Let us then divide w into $xu_1 \cdots u_p y$, where each u_i belongs to the language represented by the expression

$$001(0 \mid 1)^n 001(0 \mid 1)^n (010 \mid 100 \mid 101)[0] \cdots (010 \mid 100 \mid 101)[2^n - 1].$$

It is clear that there is only one possible to split w in this way, as each u_i begins with two $(n+3)$ -subwords with prefix 001 and contains no more $(n+3)$ -subwords with this prefix.

Since w belongs to $L_s(\mathcal{A}_1)$ it must also belong to $v_j(\mathcal{A}_1)$, for each $0 \leq j \leq 2^n - 1$. Let $\rho_0, \dots, \rho_{2^n-1}$ be accepting runs over the word w for $v_0(\mathcal{A}_1), \dots, v_{2^n-1}(\mathcal{A}_1)$, respectively. We are interested in the states assigned by the runs $\rho_0, \dots, \rho_{2^n-1}$ only to those positions of w that immediately preceded one of the subwords of w of the form u_i , for

$1 \leq i \leq p$. For simplicity, we denote each such state by $\rho_j(u_i)$; that is, $\rho_j(u_i)$ is the state that is assigned by ρ_j to the subword x , if $i = 1$, and to $xu_1 \cdots u_{i-1}$, if $1 < i \leq p$.

Consider the following sequence $d_1, \dots, d_p$ of configurations of $\mathcal{M}$: The j th position of the tape in a given d_i contains a symbol $a \in \Sigma \cup \{B\}$ if and only if $\rho_j(u_i) = q_a$. Further, it contains a symbol $a \in \Sigma \cup \{B\}$ and the head is in position j in state s if and only if $\rho_j(u_i) = q_{a,s}$. In any other case it contains the symbol B . All that is left to do is to show that each d_i is, indeed, a configuration of $\mathcal{M}$ (for instance, that the head is not assigned to two different positions of the tape by d_i), and that $d_1, \dots, d_p$ represents an accepting computation of $\mathcal{M}$ on input $\bar{a}$. This follows from the following claim.

CLAIM 10. *The following hold.*

- (1) *The configuration d_1 corresponds the initial configuration of $\mathcal{M}$ on input $\bar{a}$.*
- (2) *For each $1 \leq i < p$, d_{i+1} is a configuration of $\mathcal{M}$ that can be obtained from d_i by applying the transition rules of $\mathcal{M}$.*
- (3) *The configuration d_p is a final configuration for $\mathcal{M}$.*

PROOF. We start by proving (1): The configuration d_1 corresponds to the initial configuration of $\mathcal{M}$. That is, we need to prove the following: (i) $\rho_0(u_1) = q_{a_0, s_0}$, (ii) $\rho_j(u_1) = q_{a_j}$, for each $1 \leq j \leq k-1$, and (iii) $\rho_j(u_1) = q_B$, for each $k \leq j \leq 2^n - 1$.

We start by showing that $\rho_0(u_1) = q_{a_0, s_0}$. Since w belongs to the language defined by each NFA in $\mathbb{A}_2$, the first $2^n (n+3)$ -subwords of w must begin with the prefix 000, and, in particular, the 2^n -th $(n+3)$ -subword of w is precisely 000[$2^n - 1$]. We then know from the construction of $\mathcal{A}_1$ that $\rho_0(u_1) = q_{a_0, s_0}$ holds if and only if ρ_0 assigns state q'_0 to some position in w before reading word 000[$2^n - 1$].

Again, since w belongs to $\bigcap_{A \in \mathbb{A}_2} L(A)$, we know that the first $2^n (n+3)$ -subwords of w are of the form:

$$000[1] 000[2] \cdots 000[2^n - 1].$$

Then, notice that from the construction of $\mathcal{A}_1$, it is only possible to reach a final state of $v_0(\mathcal{A}_1)$ by using a transition labeled with 000[0] (corresponding to the transitions $v_0((q_r, 000\bar{X}, q'_r)) = (q_r, 000v_0(\bar{X}), q'_r) = (q_r, 000[0], q'_r)$ for each $0 \leq r \leq k$). But notice that the only $(n+3)$ -subword of w the form 000[0] is the very first one. We conclude that the only way of reaching a final state from q_0 is to use the $(n+3)$ -subword 000[0], present at the beginning of w , to go from state q_0 into state q'_0 . We conclude that $\rho_0(u_1) = q_{a_0, s_0}$.

That $\rho_j(u_1) = q_{a_j}$, for each $1 \leq j \leq k-1$, can be proved using essentially the same argument, together with the fact that, for each $1 < j \leq k$, $v_j(\mathcal{A}_1)$ can only reach a final state by means of the transition $(q_j, 000v_j(\bar{X}), q'_j)$.

Finally, notice that w starts with 000[0] 000[1] $\cdots$ 000[$k-1$], and thus each $v_j(\mathcal{A}_1)$, for $j \geq k$, can only advance to state q_k when reading the prefix 000[0] 000[1] $\cdots$ 000[$k-1$] of w . This proves that $\rho_j(u_1) = q_B$ for each $j \geq k$.

We continue by proving (2): For each $1 \leq i < p$, d_{i+1} is a configuration that can be obtained from d_i by applying the transition rules of $\mathcal{M}$.

For an arbitrary $i \in [1, p-1]$, assume for each $0 \leq j \leq 2^{n-1}$ that the j th cell of d_i contains the symbol $c_j \in \Sigma \cup \{B\}$, and the head of $\mathcal{M}$ is in state s_ℓ , $0 \leq \ell \leq m$, reading position t of the tape ($0 \leq t \leq 2^n - 1$). Then, by definition of d_i , $\rho_j(u_i) = q_{c_j}$, for each $0 \leq j \leq 2^{n-1}$, except for $\rho_t(u_i)$, that is equal to q_{c_t, s_ℓ} . We need to show that d_{i+1} can differ from d_i exclusively in positions $t-1$, t and $t+1$, and that this change is consistent with $\delta^{\mathcal{M}}$.

We begin by showing that the change in the t th cell is consistent with $\delta_{\mathcal{M}}$. It is easy to see from the construction of $\mathcal{A}_1$ that the only transitions that advance from a state of

form q_{c_t, s_ℓ} lead to state of form q_b , for some $b \in \Sigma \cup \{B\}$. Assume then that $\rho_t(u_{i+1}) = q_b$, for some $b \in \Sigma \cup \{B\}$. Recall that the only possible transitions leaving from state q_{c_t, s_ℓ} into q_b in $\mathcal{A}_1$ are of the form:

- $(q_{c_t, s_\ell}, 001[\ell'] 001\bar{X}(010(0 \mid 1)^n)^* 100(0 \mid 1)^n (010(0 \mid 1)^n)^*, q_b)$, or
- $(q_{c_t, s_\ell}, 001[\ell'] 001\bar{X}(010(0 \mid 1)^n)^* 101(0 \mid 1)^n (010(0 \mid 1)^n)^*, q_b)$,

for some $0 \leq \ell' \leq m$, depending on whether there are transitions in $\delta^{\mathcal{M}}$ of the form $\delta(s_\ell, c_t) = (b, s_{\ell'}, L)$ or $\delta(s_\ell, c_t) = (b, s_{\ell'}, R)$, respectively.

But since ρ_t leads, over u_i , from state q_{c_t, s_ℓ} into q_b , it must be the case that u_i belongs to either one of the languages that label these transitions (notice that it cannot belong to both since the first one allows for the existence of an $(n+3)$ subword with prefix 100, while the second one does not). Assume that the word u_i belongs to the language:

$$001[\ell'] 001\bar{X}(010(0 \mid 1)^n)^* 100(0 \mid 1)^n (010(0 \mid 1)^n)^*,$$

the other case is completely symmetrical.

Then, a transition of the form:

$$(q_{c_t, s_\ell}, 001[\ell'] 001\bar{X}(010(0 \mid 1)^n)^* 100(0 \mid 1)^n (010(0 \mid 1)^n)^*, q_b),$$

exists in $\mathcal{A}_1$. From the construction of $\mathcal{A}_1$ it must be the case that $\delta^{\mathcal{M}}$ contains a transition $\delta(s_\ell, c_t) = (b, s_{\ell'}, L)$. This shows that the change in the t th cell of $\mathcal{M}$ is consistent with $\delta_{\mathcal{M}}$.

Next we prove that the only other cell (that is, besides the t th cell) that changes from d_i to d_{i+1} is the $m-1$ -th cell. The symbol written in this cell remains unchanged from d_i (i.e., it continues being c_{t-1}), but we must reflect now the fact that in d_{i+1} the head of $\mathcal{M}$ should be reading $t-1$ in state $s_{\ell'}$.

We start by proving that the content of each cell other than t and $t-1$ remains unchanged from d_i into d_{i+1} . To that extent, notice that since w belongs to the language given by the intersection of all automata in $\mathbb{A}_4$, we can be much more precise about the form of u_i . Indeed, we can assume that u_i is of the form:

$$001[\ell'] 001[t] (010(0 \mid 1)^n)^* 100[t-1] (010(0 \mid 1)^n)^*.$$

Then, it is easy to see that each ρ_j , for $j \notin \{t, t-1\}$, can use the transition

$$(q_{c_j}, 001[\ell'] 001(0 \mid 1)^n ((010 \mid 100 \mid 101)(0 \mid 1)^n)^* 010\bar{X}((010 \mid 100 \mid 101)(0 \mid 1)^n)^*, q_{c_j})$$

to stay in the same state as before (namely, q_{c_j}). Moreover, they cannot use any of the transitions:

$$(q_{c_j}, 001[\ell'] 001(0 \mid 1)^n (010(0 \mid 1)^n)^* (100 \mid 101)\bar{X}(010(0 \mid 1)^n)^*, q_{c_j, s_{\ell'}}),$$

for $0 \leq \ell' \leq m$, as the only $(n+3)$ -subword of u_i that starts with 100 or 101 is $100[t-1]$. We conclude that each $\rho_j(u_{i+1}) = \rho_j(u_i)$, for each $j \notin \{t, t-1\}$, which entails that d_i and d_{i+1} are exactly the same in all positions except for the t -th and the $(t-1)$ -th positions.

Finally, notice that state $q_{c_{t-1}}$ cannot be reached with u_i using the self loop transition:

$$(q_{c_{t-1}}, 001[\ell'] 001(0 \mid 1)^n ((010 \mid 100 \mid 101)(0 \mid 1)^n)^* 010\bar{X}((010 \mid 100 \mid 101)(0 \mid 1)^n)^*, q_{c_{t-1}}),$$

form some $0 \leq \ell' \leq m$, because there is no $(n+3)$ subword of u_i of the form $010[t]$. Indeed, if this was the case, then u_i must contain both $010[t-1]$ and $100[t-1]$ as $(n+3)$ -subwords, which contradicts the fact that w belongs to the language given by

the intersection of all NFAs in $\mathbb{A}_3$. Thus, since u_i begins with $001[\ell']$, the only state reachable from $q_{c_{t-1}}$ in $v_{t-1}(\mathcal{A}_1)$ is $q_{c_{t-1},s_{\ell'}}$, by using the transition

$$(q_{c_{t-1}}, 001[\ell'] 001(0 \mid 1)^n (010(0 \mid 1)^n)^* (100 \mid 101)\bar{X}(010(0 \mid 1)^n)^*, q_{c_{t-1},s_{\ell'}}).$$

We then have that $\rho_{t-1}(u_{i+1}) = q_{c_{t-1},s_{\ell'}}$. This proves that the $(t-1)$ -th symbol in d_{i+1} is equal to c_{t-1} , and that the head is in the $(t-1)$ -th position in state $s_{\ell'}$. Further, the head is in no other position in d_{i+1} .

We finally prove (3): d_p is a final configuration for $\mathcal{M}$. This follows easily from the fact that d_p must be a valid configuration of $\mathcal{M}$, no transition is given from state q_f , and the only states that can reach the final state q_f of $\mathcal{A}$ are states of the form q_a or q_{a,s_f} , for $a \in \Sigma \cup \{B\}$. This finishes the proof of the claim. $\square$

($\Leftarrow$) Assume, on the other hand, that there is an accepting computation $d_1, \dots, d_p$ of $\mathcal{M}$ on input $\bar{a}$. That is, $d_1, \dots, d_p$ is a sequence of configurations of $\mathcal{M}$ such that: (1) d_1 is the initial configuration of $\mathcal{M}$ on input $\bar{a}$; (2) d_p is a final configuration for $\mathcal{M}$; and (3) for each $1 \leq i < p$, d_{i+1} is a configuration that can be obtained from d_i by applying the transition rules of $\mathcal{M}$. We show that $L_s(\mathcal{A}_1) \cap \bigcap_{A \in \mathbb{A}_2, \mathbb{A}_3, \mathbb{A}_4} L(A)$ is nonempty.

Construct a word $w = xu_1 \dots u_p y$ over Σ as follows.

- $x = 000[0] \dots 000[2^n - 1]$.
- Assume that in the final configuration d_p the head of $\mathcal{M}$ is in position j ($1 \leq j \leq 2^n - 1$). Then, $y := 111[j]$.
- The word u_i , for $1 \leq i \leq p$, is constructed as follows. Assume that in d_i , $1 \leq i \leq p$, $\mathcal{M}$ is in state s_j , $0 \leq j \leq m$, and with its head pointing to the ℓ th position of the tape ($0 \leq \ell \leq 2^n - 1$). Further, assume that in d_i the ℓ 'th cell of the tape contains the symbol $c_{\ell'} \in \Sigma \cup \{B\}$, for each $0 \leq \ell' \leq 2^n - 1$, and that d_{i+1} is obtained from d_i by applying a transition of the form $\delta(s_j, a) = (s_{j'}, b, D)$, where $0 \leq j' \leq m$, $a \in \Sigma \cup \{B\}$, $b \in \Sigma$, and D is either L or R . Then, u_i is:
 - $001[\ell] 001[j'] 010[0] \dots 010[\ell - 1] 010[\ell] \dots 010[2^n - 1]$, if $D = L$, and
 - $001[\ell] 001[j'] 010[0] \dots 010[\ell] 101[\ell + 1] 010[\ell + 2] \dots 010[2^n - 1]$, if $D = R$.

We prove next that $w \in L_s(\mathcal{A}_1) \cap \bigcap_{A \in \mathbb{A}_2, \mathbb{A}_3, \mathbb{A}_4} L(A)$. We only give the proof for $\mathcal{A}_1$, as the rest is straightforward. We split the proof in several parts.

CLAIM 11. *The following hold.*

- (1) State q_{a_0,s_0} is reachable from q_0 in $v_0(\mathcal{A}_1)$ using x .
- (2) For each $1 \leq j \leq k-1$, state q_{a_j} is reachable from q_0 in $v_j(\mathcal{A}_1)$ using x .
- (3) For each $k \leq j \leq 2^n - 1$, state q_B is reachable from q_0 in $v_j(\mathcal{A}_1)$ using x .

PROOF. For the first statement we run the automaton $v_0(\mathcal{A}_1)$ as follows. Start in q_0 , then, by reading $000[0]$ advance to q'_0 using transition $(q_0, 0000^n, q'_0)$, then loop in q'_0 using transition $(q'_0, 000(0 \mid 1)^n, q'_0)$ until the $(n+3)$ subword 0001^n is reached, in which case we advance to q_{a_0,s_0} using transition $(q'_0, 0001^n, q_{a_0,s_0})$.

For the second statement, let $j \in [1, k-1]$. Then, run $v_j(\mathcal{A}_1)$ as follows. Advance from q_0 to q_j using words $000[0]$ to $000[j-1]$ and transitions

$$(q_0, 000(0 \mid 1)^n, q_1)(q_1, 000(0 \mid 1)^n, q_2) \dots (q_{j-1}, 000(0 \mid 1)^n, q_j).$$

Then, move from q_j to q'_j using word $000[j]$ and transition $(q_j, 000[j], q'_j)$ (recall that $v_j(\bar{X}) = [j]$), loop in q'_j using transition $(q'_j, 000(0 \mid 1)^n, q'_j)$ until the $(n+3)$ subword 0001^n is reached, in which case we advance to q_{a_j} using transition $(q'_j, 0001^n, q_{a_j})$.

For the last statement, let $j \in [k, 2^n - 1]$. Starting from q_0 in $v_j(\mathcal{A}_1)$, advance from q_0 to q_k (as in the previous case) until the $(n + 3)$ subword $000[j]$ is reached, then if $j \neq 2^n - 1$ move into state q'_k using word $000[j]$ and transition $(q_k, 000[j], q'_k)$, looping in q'_k until word 0001^n is seen, in which we case we move into q_B . If $j = 2^n - 1$ move directly from q_k into q_B using transition $(q_k, 0001^n, q_B)$ and word 0001^n . $\square$

CLAIM 12. *Assume that in configuration d_i , $1 \leq i \leq p$, $\mathcal{M}$ is in state s_ℓ , $0 \leq \ell \leq m$, with the head pointing to the t th position of the tape ($0 \leq t \leq 2^n - 1$), and the j th cell of the tape contains symbol $c_j \in \Sigma \cup \{B\}$, for each $0 \leq j \leq 2^n - 1$. Further, d_{i+1} is obtained from $\mathcal{M}$ by applying a transition of the form $\delta(s_\ell, c_t) = (s_{\ell'}, b, D)$, where $0 \leq \ell' \leq m$, $b \in \Sigma$ and D is either L or R . Then the following hold.*

- (1) *State q_b is reachable from $q_{c_t, s}$ in $v_t(\mathcal{A}_1)$ using u_i .*
- (2) *If $D = L$, for each $j \notin \{t, t - 1\}$, state q_{c_j} is reachable from q_{c_j} in $v_j(\mathcal{A}_1)$ using u_i .*
- (3) *If $D = R$, for each $j \notin \{t, t + 1\}$, state q_{c_j} is reachable from q_{c_j} in $v_j(\mathcal{A}_1)$ using u_i .*
- (4) *If $D = L$, then state $q_{b, s'}$ is reachable from $q_{c_{t-1}}$ in $v_{t-1}(\mathcal{A}_1)$ using u_i .*
- (5) *If $D = R$, then state $q_{b, s'}$ is reachable from $q_{c_{t+1}}$ in $v_{t+1}(\mathcal{A}_1)$ using u_i .*

PROOF. Assume that $D = L$ (the other case is completely symmetric). Then, by definition of w , u_i is of the form:

$$001[t] 001[\ell'] 010[0] \dots 010[t - 2] 100[t - 1] 010[t] \dots 010[2^n - 1].$$

Then clearly, the following hold:

- (1) The automaton $v_t(\mathcal{A}_1)$ can reach q_b from q_{c_t, s_ℓ} using

$$(q_{c_t, s_\ell}, 001[\ell'] [t] (010(0 | 1)^n)^* 100(0 | 1)^n (010(0 | 1)^n)^*, q_b).$$

- (2) Each $v_j(\mathcal{A}_1)$, $j \notin \{t, t - 1\}$ can use transition

$$(q_{c_j}, 001[\ell'] 001(0 | 1)^n (010 | 100 | 101)(0 | 1)^n)^* 010[j] ((010 | 100 | 101)(0 | 1)^n)^*, q_{c_j})$$

to stay in state q_{c_j} .

- (3) Finally, $v_{t-1}(\mathcal{A}_1)$ uses transition

$$(q_{c_{t-1}}, 001[\ell'] 001(0 | 1)^n (010(0 | 1)^n)^* (100 | 101)[t - 1] (010(0 | 1)^n)^*, q_{c_{t-1}, s_{\ell'}})$$

to advance from $q_{c_{t-1}}$ to $q_{b, s_{\ell'}}$. $\square$

CLAIM 13. *The following hold.*

- (1) *State q_f is reachable from q_a using y , for each $a \in \Sigma \cup \{B\}$.*
- (2) *Assume that the machine halts with the head in the t th position of the tape ($0 \leq t \leq 2^n - 1$). Then state q_f is reachable from q_{a, s_m} in $v_t(\mathcal{A}_1)$ using y , for each $a \in \Sigma \cup \{B\}$.*

PROOF. Easy since $\mathcal{A}_1$ contains transitions $(q_a, 111(0 | 1)^n, q_f)$, for every $a \in \Sigma \cup \{B\}$; and $(q_{a, s_m}, 111\bar{X}, q_f)$ for every $a \in \Sigma \cup \{B\}$. $\square$

Using these three claims and the fact that $d_1, \dots, d_p$ is valid computation for $\mathcal{M}$ on input $\bar{a}$, it is now easy to prove that $v_j(\mathcal{A}_1)$ accepts w , for each $0 \leq j \leq 2^n - 1$. This finishes the proof of the theorem. $\square$

Proof of Theorem 6.8 using Codd Wildcard Automata

In the proof of Theorem 6.8, we constructed, given a turing machine $\mathcal{M}$ and a word $\bar{a}$ in $\{0, 1\}^*$, a wildcard automaton $\mathcal{A}_1$ and sets $\mathbb{A}_2, \mathbb{A}_3, \mathbb{A}_4$ of NFAs such that $L_s(\mathcal{A}_1) \cap \bigcap_{A \in \mathbb{A}_2, \mathbb{A}_3, \mathbb{A}_4} L(A)$ is not empty if and only if $\mathcal{M}$ accepts on input $\bar{a}$.

We now briefly discuss how to modify this reduction so that $\mathcal{A}_1$ uses each variable exactly once. More precisely, we describe how to construct an incomplete automaton $\mathcal{B}_1$ and sets $\mathbb{B}_2, \mathbb{B}_3, \mathbb{B}_4, \mathbb{B}_5$ of NFAs such that $\mathcal{B}_1$ uses each variable only once in its transitions, and $L_s(\mathcal{B}_1) \cap \bigcap_{A \in \mathbb{B}_2, \mathbb{B}_3, \mathbb{B}_4, \mathbb{B}_5} L(A)$ is not empty if and only if $\mathcal{M}$ accepts on input $\bar{a}$.

We begin by recalling some definitions and facts from the proof of Theorem 6.8. Recall that we assumed for convenience that the transitions of $\mathcal{A}_1$ could be defined using words, that is, they were defined over $Q \times (\Sigma \cup \mathcal{W})^* \times Q$. In this proof it is convenient to work instead with an equivalent automaton $\mathcal{A}'_1$ whose transitions are defined over $Q \times (\Sigma \cup \mathcal{W}) \times Q$, that is, over symbols of $\Sigma \cup \mathcal{W}$. Clearly this automata can be constructed in polynomial time from $\mathcal{A}_1$. Assume that $\mathcal{A}'_1$ is of the form $\mathcal{A}'_1 = (Q', \Sigma, \mathcal{W}, q'_0, F', \delta')$, and note also that the label variables of $\mathcal{A}'_1$ are the same than those of $\mathcal{A}_1$, and more precisely, $\mathcal{W} = \{X_1, \dots, X_n\}$. From the construction of the automaton $\mathcal{A}_1$ in the previous proof, we can assume that all of the transitions in δ' that are labelled with a variable from $\mathcal{W}$ are of the following form $(\dagger)$:

and where no other transition was defined for any of the states $\{q_1, \dots, q_n\}$. Finally, in this proof we let $[q]$ denote the unary codification of a given state $q \in Q'$.

Next we continue with the description of automata $\mathcal{B}_1$ and $\mathbb{B}_2, \mathbb{B}_3, \mathbb{B}_4, \mathbb{B}_5$. Let us begin with the construction of $\mathcal{B}_1$.

Consider first the incomplete automaton $\mathcal{B}'_1$ defined as

$$\mathcal{B}'_1 = (Q' \cup \{p_1^X, \dots, p_{n+1}^X\}, \Sigma, \mathcal{W}, q'_0, F', \delta_{\mathcal{B}}),$$

where $\delta_{\mathcal{B}}$ is defined as follows:

- $\delta_{\mathcal{B}}$ contains transitions $(p_1^X, X_1, p_2^X), (p_2^X, X_2, p_3^X), \dots, (p_n^X, X_n, p_{n+1}^X)$;
- if $(q, a, q') \in \delta'$, for some $a \notin \mathcal{W}$, then $(q, a, q') \in \delta_{\mathcal{B}}$; and
- For every set of transitions in δ' that are labelled with $\bar{X}$ (i.e., every set of n transitions of the form $(\dagger)$, for some set of states $\{q_1, \dots, q_{n+1}\}$ in Q'), add the transitions $(q_1, [q_{n+1}], p_1^X)$ and $(p_{n+1}^X, [q_{n+1}], q_{n+1})$ to $\delta_{\mathcal{B}}$.

Note that $\mathcal{B}'_1$ uses variables $X_1, \dots, X_n$ in a single transition in $\delta_{\mathcal{B}}$, and that this construction can clearly be performed in polynomial time.

The intuition behind this construction is the following.

Since $\mathcal{B}'_1$ can only use variables only once, all the transitions that used $\bar{X}$ in $\mathcal{A}'_1$ need now to go through the same set $(p_1^X, X_1, p_2^X), (p_2^X, X_2, p_3^X), \dots, (p_n^X, X_n, p_{n+1}^X)$ of transitions.

The idea is to direct all of the transitions in $\mathcal{A}'_1$ that used variables $X_1, \dots, X_n$ onto the single path that starts in state p_1^X and ends in p_{n+1}^X . Of course, since NFAs have no memory, if we do this in the naive way we automatically lose the information of which particular transition in $\mathcal{A}'_1$ we were using. Thus, we need to *remember* to which of the transitions of $\mathcal{A}_1$ we were using. This is achieved by the extra bits used to code states

$q \in Q$ as a binary string $[q]$. For example, if for states $\{q_1, \dots, q_{n+1}\}$ and $\{q'_1, q'_{n+1}\}$ in $\mathcal{A}'_1$, one has transitions of form.

Then we shall transform them into transitions over $\mathcal{B}'_1$ as follows.

Our only problem now is how to tell the transitions to remember where they were originally going in $\mathcal{A}_1$, that is, is one starts in q_1 then one can only go to q_{n+1} , and if one starts in q'_1 then one can only go to q'_{n+1} . We can achieve this by adding to our set of NFAs an additional set of automata that ensures that, whenever one goes to this types of transitions, then the incoming and outgoing states should be the same. In other words, if c is the number of states in $\mathcal{A}'_1$, we now need to accept words of form $((0|1)^{n+2c+3})^*$, where each such $(n + 2c + 3)$ -subword of this expression matches $(0|1)^3[q](0|1)^n[q]$, for some $q \in Q$. This set can clearly be constructed in polynomial time using the same techniques as the previous version of the reduction.

Denote then by $\mathbb{B}_5$ the sets of automata such that their intersection accepts such language. In addition, let us define $\mathbb{B}_2, \mathbb{B}_3, \mathbb{B}_4$ as the automaton obtained by modifying the sets $\mathbb{A}_2, \mathbb{A}_3, \mathbb{A}_4$ so that they now work under our coding of $(n + 2c + 3)$ -subwords (instead of $(n + 1)$ -subwords).

Finally, we need to modify the rest of the transitions of $\mathcal{B}'_1$ so that they work under this modified coding (e.g., those corresponding to the initial state). But it is not difficult to define them using the techniques presented in the proof of Theorem 6.8. We denote the resulting automata by $\mathcal{B}_1$. This finishes our description of automata $\mathbb{B}_2, \mathbb{B}_3, \mathbb{B}_4, \mathbb{B}_5$, and $\mathcal{B}_1$.

Proof of Claim 7

Let $s_1, \dots, s_p$ be an arbitrary enumeration of the states in S , f be a mapping from S to 2^S , $S' \subseteq S$ and $\mathcal{G} \subseteq 2^S$. In order to determine whether $(f, S', \mathcal{G})$ is realized in $\mathcal{A}_2$, we do the following. First, construct in constant time a deterministic NFA $\mathcal{A}_3$ that is equivalent to $\mathcal{A}_1$ using the standard powerset construction. Thus, the states of $\mathcal{A}_3$ are precisely the subsets S' of S . Assume that $\delta' : 2^S \times \Sigma \rightarrow 2^S$ is the transition function of $\mathcal{A}_3$. Then, from $\mathcal{A}_3$ we construct, in constant time, two deterministic NFAs $\mathcal{A}_4$ (without initial/final states) and $\mathcal{A}_5$ as follows.

- The set of states of $\mathcal{A}_4$ is the disjoint union between $\{S' \mid S' \subseteq S\}$ and $\{S' \cup \{s_{aux}\} \mid S' \subseteq S\}$, where s_{aux} is a new auxiliary state. The transition function of $\mathcal{A}_4$ is constructed from the transition function δ' of $\mathcal{A}_3$ using the following rules: (1) If

transition from $S'_1 \subseteq S$ into $S'_2 \subseteq S$ labeled a exists in $\mathcal{A}_3$, add the same transition to $\mathcal{A}_4$, and add also a transition labeled a from $S'_1 \cup \{s_{aux}\}$ into $S'_2 \cup \{s_{aux}\}$; (2) Delete every transition labeled a from a state $S'_1 \subseteq S$, such that $S'_1 \cap F = \emptyset$, into a state $S'_2 \subseteq S$ such that $S'_2 \cap F \neq \emptyset$, and replace it by a transition labeled a from S'_1 into $S'_2 \cup \{s_{aux}\}$.

Intuitively, s_{aux} works as a flag for states of $\mathcal{A}_3$ that contain some final state in F . Thus, the transition function of $\mathcal{A}_4$ leads from a state $\{s\}$ to a state $f(s) \cup \{s_{aux}\}$ over a word w if and only if $\delta'(\{s\}, w) = f(s)$, and there is a prefix w' of w such that $\delta'(\{s\}, w')$ contains some state in F . On the contrary, the transition function of $\mathcal{A}_4$ leads from a state $\{s\}$ to a state $f(s)$ over a word w if and only if $\delta'(\{s\}, w) = f(s)$, and there is no prefix w' of w such that $\delta'(\{s\}, w')$ contains some state in F .

- The set of states of $\mathcal{A}_5$ is 2^{2^S} . Its initial state is $\{\{\}\}$ and its final state is $\mathfrak{S}$. There is a transition labeled a from $\{S_1, \dots, S_\ell\}$ into $\{S'_1, \dots, S'_{\ell'}\}$ if and only if $\{S'_1, \dots, S'_{\ell'}\}$ is precisely the set that contains (1) all states of $\mathcal{A}_3$ of the form $\delta'(S_i, a)$, for $1 \leq i \leq \ell$, and (2) state $\delta(\{s_0\}, a)$.

Intuitively, the transition function of $\mathcal{A}_5$ leads from state $\{\{\}\}$ into state $\mathfrak{S}$ over a word w if and only if the set $\mathfrak{S}$ consists of exactly those states S'' of $\mathcal{A}_3$ such that for some suffix w' of w it is the case that $\delta'(\{s_0\}, w') = S''$.

Then, we construct, in constant time, p copies $\mathcal{A}_4^1, \dots, \mathcal{A}_4^p$ of $\mathcal{A}_4$, and set the initial state of $\mathcal{A}_4^i$ ($1 \leq i \leq p$) to be $\{s_i\}$ and its final state to be $f(s_i) \cup \{s_{aux}\}$, if $s_i \in S'$, and $f(s_i)$ otherwise. Afterwards, we construct in polynomial time the NFA $\mathcal{A}_6 := \mathcal{A}_2 \times \mathcal{A}_4^1 \times \dots \times \mathcal{A}_4^p \times \mathcal{A}_5$. Clearly, the tuple $(f, S', \mathfrak{S})$ is realized in $\mathcal{A}_1$ if and only if $\mathcal{A}_6$ is nonempty. But the latter can be checked in polynomial time in the size of $\mathcal{A}_6$, and, thus, in the size of $\mathcal{A}_2$. Now the claim follows from the fact that there is a constant number of tuples of the form $(f, S', \mathfrak{S})$. $\square$