P-Optimal Proof Systems for Each NP-Complete Set
but no Complete Disjoint NP-Pairs Relative to an Oracle

Titus Dose
Julius-Maximilians-Universität Würzburg
10th January 2020

Abstract
Pudlák [Pud17] lists several major conjectures from the field of proof complexity and asks
for oracles that separate corresponding relativized conjectures. Among these conjectures are:

- **DisjNP**: The class of all disjoint NP-pairs does not have many-one complete elements.
- **SAT**: NP does not contain many-one complete sets that have P-optimal proof systems.
- **UP**: UP does not have many-one complete problems.
- **NP ∩ coNP**: NP ∩ coNP does not have many-one complete problems.

As one answer to this question, we construct an oracle relative to which DisjNP, ¬SAT, UP, and NP ∩ coNP hold, i.e., there is no relativizable proof for the implication DisjNP ∧ UP ∧ NP ∩ coNP ⇒ SAT. In particular, regarding the conjectures by Pudlák this extends a result by Khaniki [Kha19].

1 Introduction

The main motivation for the present paper is an article by Pudlák [Pud17] that is “motivated by
the problem of finding finite versions of classical incompleteness theorems”, investigates major
conjectures in the field of proof complexity, discusses their relations, and in particular draws
new connections between the conjectures. Among others, Pudlák conjectures the following
assertions (note that within the present paper all reductions are polynomial-time-bounded):

- **CON** (resp., SAT): coNP (resp., NP) does not contain many-one complete sets that have
  P-optimal proof systems
- **CON^N**: coNP does not contain many-one complete sets that have optimal proof systems,
  (note that CON^N is the non-uniform version of CON)
- **DisjNP** (resp., DisjCoNP): The class of all disjoint NP-pairs (resp., coNP-pairs) does not
  have many-one complete elements,
- **TFNP**: The class of all total polynomial search problems does not have complete elements,
- **NP ∩ coNP** (resp., UP): NP ∩ coNP (resp., UP, the class of problems accepted by NP ma-
  chines with at most one accepting path for each input) does not have many-one complete
  elements.

The following figure contains the conjectures by Pudlák and illustrates the state of the art
regarding (i) known implications and (ii) separations in terms of oracles that prove the non-
existence of relativizable proofs for implications. O denotes the oracle constructed in the present
paper.
The main conjectures of [Pud17] are CON and TFNP. Let us give some background on these conjectures (for details we refer to [Pud13]) and on the notion of disjoint pairs. The first main conjecture CON refers to the notion of proof systems introduced by Cook and Reckhow [CR79], who define a proof system for a set \( A \) to be a polynomial-time computable function whose range is \( A \).

The subsequent paragraph is due to [DG19] and explains a logical characterization of CON and CON\(^N\). CON has an interesting connection to some finite version of an incompleteness statement. Denote by \( \text{Con}_T(n) \) the finite consistency of a finitely axiomatized theory \( T \), i.e., \( \text{Con}_T(n) \) is the statement that \( T \) has no proofs of contradiction of length \( \leq n \). Krajíček and Pudlák [KP89] raise the conjectures CON and CON\(^N\) and show that the latter is equivalent to the statement that there is no finitely axiomatized theory \( S \) which proves the finite consistency \( \text{Con}_T(n) \) for every finitely axiomatized theory \( T \) by a proof of polynomial length in \( n \). In other words, \( \neg\text{CON} \) expresses that a weak version of Hilbert’s program (to prove the consistency of all mathematical theories) is possible [Pud96]. Correspondingly, \( \neg\text{CON} \) is equivalent to the existence of a theory \( S \) such that, for each fixed finitely axiomatized theory \( T \), proofs of \( \text{Con}_T(n) \) in \( S \) can be constructed in polynomial time in \( n \) [KP89].

The conjecture TFNP, raised by Megiddo and Papadimitriou [MP91], is implied by the non-existence of disjoint coNP-pairs [BKM09, Pud17], and implies that no NP-complete set has P-optimal proof systems [BKM09, Pud17]. It states the non-existence of total polynomial search problems that are complete with respect to polynomial reductions, where a total polynomial
search problem (i) is represented by a polynomial $p$ and a binary relation $R$ satisfying $\forall x \exists y |y| \leq p(|x|) \land (x, y) \in R$ and (ii) is the following computational task: On input $x$ compute some $y$ with $|y| \leq p(|x|) \land (x, y) \in R$. In other words, total polynomial search problems are represented by nondeterministic multivalued functions with values that are polynomially verifiable and guaranteed to exist [MP91].

The notion of disjoint NP-pairs, i.e., pairs $(A, B)$ with $A \cap B = \emptyset$ and $A, B \in \text{NP}$, has its origin in public-key cryptography and characterizes promise problems [EY80, ESY84, GS88]. Razborov [Raz94] connects disjoint pairs with the concept of propositional proof systems (pps), i.e., proof systems for the set of propositional tautologies TAUT, defining for each pps $f$ a disjoint NP-pair, the so-called canonical pair of $f$, and showing that the canonical pair of a P-optimal pps $f$ is complete. Hence, putting it another way, DisjNP $\Rightarrow$ CON, which Köbler, Messner, and Torán [KMT03] extend to DisjNP $\Rightarrow$ CONN.

In contrast to the many implications only very few oracles were known separating two of the relativized conjectures [Pud17], which is why Pudlák asks for further oracles showing relativized conjectures to be different.

Khaniki [Kha19] partially answers this question: besides showing two of the conjectures to be equivalent he presents two oracles $\mathcal{V}$ and $\mathcal{W}$ showing that SAT and CON (as well as TFNP and CON) are independent in relativized worlds which means that none of the two possible implications between the two conjectures has a relativizable proof. To be more precise, relative to $\mathcal{V}$, there exist P-optimal propositional proof systems but no many-one complete disjoint coNP-pairs, where —as mentioned above— the latter implies TFNP and SAT. Relative to $\mathcal{W}$, there exist no P-optimal propositional proof systems and each total polynomial search problem has a polynomial-time solution, where the latter implies $\neg$SAT [KM00].

Dose and Glaßer [DG19] construct an oracle $X$ that also separates some of the above relativized conjectures. Relative to $X$ there exist no many-one complete disjoint NP-pairs, UP has many-one complete problems, and NP $\cap$ coNP has no many-one complete problems. In particular, relative to $X$, there do not exist P-optimal propositional proof systems. Thus, among others, $X$ shows that the conjectures CON and UP as well as NP $\cap$ coNP and UP cannot be proven equivalent with relativizable proofs.

Our Contribution. In the present paper we construct an oracle $O$ relative to which

1. The class of all disjoint NP-pairs does not have many-one complete elements.
2. Each many-one complete set for NP has P-optimal proof systems.
3. UP does not contain many-one complete problems.
4. NP $\cap$ coNP does not contain many-one complete problems.

Indeed, relative to $O$ there even exist no disjoint NP-pairs that are hard for NP $\cap$ coNP, which implies both 1 and 4. Figure 1 illustrates that $O$ yields one of the strongest oracle results that Pudlák [Pud17] asks for since DisjNP, UP, and NP $\cap$ coNP are the strongest conjectures in their respective branches in Figure 1 whereas SAT is the weakest conjecture that is not relativizable implied by the three other conjectures.

Among others, the oracle shows that there are no relativizable proofs for the implications NP $\cap$ coNP $\Rightarrow$ SAT and UP $\Rightarrow$ SAT. Let us now focus on the properties 1 and 2 of the oracle. Regarding these, our oracle has similar properties as the aforementioned oracle $\mathcal{W}$ by Khaniki [Kha19]: both oracles show that there is no relativizable proof for the implication CON $\Rightarrow$ SAT. Relative to Khaniki’s oracle $\mathcal{W}$ it even holds that each total polynomial search problem has a polynomial time solution, which implies not only $\neg$SAT but also that all optimal proof systems

3
for SAT are P-optimal [KM00]. Regarding Pudlák’s conjectures, however, our oracle O extends Khaniki’s result as relative to O we have the even stronger result that there is no relativizable proof for the implication DisjNP ⇒ SAT. Since due to the oracle V by Khaniki [Kha19] none of the implications DisjCoNP ⇒ DisjNP, TFNP ⇒ DisjNP, and SAT ⇒ DisjNP can be proven relativizably, our oracle shows that DisjNP is independent of each of the conjectures DisjCoNP, TFNP, and SAT in relativized worlds, i.e., none of the six possible implications has a relativizable proof.

2 Preliminaries

Throughout this paper let Σ be the alphabet {0, 1}. We denote the length of a word w ∈ Σ* by |w|. Let Σ∗<n = {w ∈ Σ* | |w| < n} for <∈ {≤, <, =, >, ≥}. The empty word is denoted by ε and the i-th letter of a word w for 0 ≤ i < |w| is denoted by w(i), i.e., w = w(0)w(1) · · · w(|w| − 1). If v is a prefix of w, i.e., |v| ≤ |w| and w(i) = v(i) for all 0 ≤ i < |v|, then we write v ⊑ w or w ⊒ v. If v ⊑ w and |v| < |w|, then we write v ⊂ w or w ⊀ v. For each finite set Y ⊆ Σ*, let ℓ(Y) ≝ ∑w∈Y |w|.

Z denotes the set of integers, N denotes the set of natural numbers, and N+ = N − {0}. The set of primes is denoted by P = {2, 3, 5, . . . } and P≥3 denotes the set P − {2}. Moreover, P1 (resp., P3) denotes the set of all primes of the form 4k + 1 (resp., 4k + 3) for k ∈ N.

We identify Σ* with N via the polynomial-time computable, polynomial-time invertible bijection w ↦⇒ ∑i<|w|(1 + w(i))2|w|−1−i, which is a variant of the dyadic encoding. Hence, notations, relations, and operations for Σ* are transferred to N and vice versa. In particular, |n| denotes the length of n ∈ N. We eliminate the ambiguity of the expressions 0i and 1i by always interpreting them over Σ*.

Let ⟨⟩ : U≥0 Nn → N be an injective, polynomial-time computable, polynomial-time invertible pairing function such that |⟨u1, . . . , un⟩| = 2(|u1| + · · · + |un| + n).

Given two sets A and B, A − B denotes the set difference between A and B, i.e., A − B = {a ∈ A | a /∈ B}. The complement of a set A relative to the universe U is denoted by A = U − A.

The domain and range of a function t are denoted by dom(t) and ran(t), respectively.

FP, P, and NP denote standard complexity classes [Pap94]. Define coC = {A ⊆ Σ* | A ∈ C} for a class C. UP is the class of all problems accepted by nondeterministic polynomial-time Turing machines that on each input have at most one accepting path. If A, B ∈ NP (resp., A, B ∈ coNP) and A ∩ B = ∅, then we call (A, B) a disjoint NP-pair (resp., a disjoint coNP-pair).

The set of all disjoint NP-pairs (resp., coNP-pairs) is denoted by DisjNP (resp., DisjCoNP).

We also consider all these complexity classes in the presence of an oracle D and denote the corresponding classes by FP D, P D, NP D, and so on.

Let M be a Turing machine. MD(x) denotes the computation of M on input x with D as an oracle. For an arbitrary oracle D we let MD = {x | MD(x) accepts}, where as usual in case M is nondeterministic, the computation MD(x) accepts if and only if it has at least one accepting path.

For a deterministic polynomial-time Turing transducer (i.e., a Turing machine computing a function), depending on the context, FD(x) either denotes the computation of F on input x with D as an oracle or the output of this computation.

Definition 2.1 A sequence (Mi)i∈N+ is called standard enumeration of nondeterministic, polynomial-time oracle Turing machines, if it has the following properties:

1. All Mi are nondeterministic, polynomial-time oracle Turing machines.
2. For all oracles $D$ and all inputs $x$ the computation $M^D_i(x)$ stops within $|x|^i + i$ steps.

3. For every nondeterministic, polynomial-time oracle Turing machine $M$ there exist infinitely many $i \in \mathbb{N}$ such that for all oracles $D$ it holds that $L(M^D) = L(M^D_i)$.

4. There exists a nondeterministic, polynomial-time oracle Turing machine $M$ such that for all oracles $D$ and all inputs $x$ it holds that $M^D((i,x,0^{|x|^i+i}))$ nondeterministically simulates the computation $M^D_i(x)$.

Analogously we define standard enumerations of deterministic, polynomial-time oracle Turing transducers.

Throughout this paper, we fix some standard enumerations. Let $M_1, M_2, \ldots$ be a standard enumeration of nondeterministic polynomial-time oracle Turing machines. Then for every oracle $D$, the sequence $(M_i)_{i \in \mathbb{N}^+}$ represents an enumeration of the languages in $\text{NP}^D$, i.e., $\text{NP}^D = \{L(M^D_i) \mid i \in \mathbb{N}\}$. Let $F_1, F_2, \ldots$ be a standard enumeration of polynomial-time oracle Turing transducers.

By the properties of standard enumerations, for each oracle $D$ the problem

$$K^D = \{(0^i, 0^i, x) \mid i, t, x \in \mathbb{N}, i > 0, \text{ and } M^D_i(x) \text{ accepts within } t \text{ steps}\}$$

is $\text{NP}^D$-complete (in particular it is in $\text{NP}^D$).

In the present article we use polynomial-time-bounded many-one reductions. Let $D$ be an oracle. For problems $A, B \subseteq \Sigma^*$ we write $A \leq_m^D B$ (resp., $A \leq_p^D B$) if there exists $f \in \text{FP}^D$ (resp., $f \in \text{FP}^D$) with $\forall x \in \Sigma^* x \in A \iff f(x) \in B$. In this case we say that $A$ is polynomially many-one reducible to $B$. Now let $A, B, A', B' \subseteq \Sigma^*$ such that $A \cap B = A' \cap B' = \emptyset$. In this paper we always use the following reducibility for disjoint pairs [Raz94]. $(A', B')$ is polynomially many-one reducible to $(A, B)$, denoted by $(A', B') \leq_{m}^{\text{NP}^D} (A, B)$, if there exists $f \in \text{FP}^D$ with $f(A') \subseteq A$ and $f(B') \subseteq B$. If $A' = \overline{B'}$, then we also write $A' \leq_{m}^{\text{NP}^D} (A, B)$ instead of $(A', B') \leq_{m}^{\text{NP}^D} (A, B)$.

We say that $(A, B)$ is $\leq_{m}^{\text{NP}^D}$-hard ($\leq_{m}^{\text{NP}^D}$-complete) for $\text{DisjNP}^D$ if $(A', B') \leq_{m}^{\text{NP}^D} (A, B)$ for all $(A', B') \in \text{DisjNP}^D$ (and $(A, B) \in \text{DisjNP}^D$). Moreover, a pair $(A, B)$ is $\leq_{m}^{\text{NP}^D}$-hard for $\text{NP}^D \cap \text{coNP}^D$ if $A' \leq_{m}^{\text{NP}^D} (A, B)$ for every $A \in \text{NP}^D \cap \text{coNP}^D$.

**Definition 2.2 ([CR79])** A function $f \in \text{FP}$ is called proof system for the set $\text{ran}(f)$. For $f, g \in \text{FP}$ we say that $f$ is simulated by $g$ (resp., $f$ is $P$-simulated by $g$) denoted by $f \leq g$ (resp., $f \leq_P^g$), if there exists a function $\pi$ (resp., a function $\pi \in \text{FP}$) and a polynomial $p$ such that $|\pi(x)| \leq p(|x|)$ and $g(\pi(x)) = f(x)$ for all $x$. A function $g \in \text{FP}$ is optimal (resp., $P$-optimal), if $f \leq g$ (resp., $f \leq_P^g$) for all $f \in \text{FP}$ with $\text{ran}(f) = \text{ran}(g)$. Corresponding relativized notions are obtained by using $P^D, \text{FP}^D$, and $\leq_P^D$ in the definitions above.

The following proposition states the relativized version of a result by Köbler, Messner, and Torán [KMT03], which they show with a relativizable proof.

**Proposition 2.3 ([KMT03])** For every oracle $D$, if $A$ has a $P^D$-optimal (resp., optimal) proof system and $B \leq_{m}^{P^D} A$, then $B$ has a $P^D$-optimal (resp., optimal) proof system.

**Corollary 2.4** For every oracle $D$, if there exists a $\leq_{m}^{P^D}$-complete $A \in \text{NP}^D$ that has a $P^D$-optimal (resp., optimal) proof system, then all sets in $\text{NP}^D$ have $P^D$-optimal (resp., optimal) proof systems.
Let us introduce some (partially quite specific) notations that are designed for the construction of oracles \cite{DG19}. The support $\text{supp}(t)$ of a real-valued function $t$ is the subset of the domain that consists of all values that $t$ does not map to 0. We say that a partial function $t$ is injective on its support if $t(i, j) = t(i', j')$ for $(i, j), (i', j') \in \text{supp}(t)$ implies $(i, j) = (i', j')$. If a partial function $t$ is not defined at point $x$, then $t \cup \{x \mapsto y\}$ denotes the extension of $t$ that at $x$ has value $y$.

If $A$ is a set, then $A(x)$ denotes the characteristic function at point $x$, i.e., $A(x) = 1$ if $x \in A$, and 0 otherwise. An oracle $D \subseteq \mathbb{N}$ is identified with its characteristic sequence $D(0)D(1)\cdots$, which is an $\omega$-word. In this way, $D(i)$ denotes both, the characteristic function at point $i$ and the $i$-th letter of the characteristic sequence, which are the same. A finite word $w$ describes an oracle that is partially defined, i.e., only defined for natural numbers $x < |w|$. We can use $w$ instead of the set $\{i \mid w(i) = 1\}$ and write for example $A = w \cup B$, where $A$ and $B$ are sets. For nondeterministic oracle Turing machines $M$ we use the following phrases: a computation $M^w(x)$ definitely accepts, if it contains a path that accepts and all queries on this path are $< |w|$. A computation $M^w(x)$ definitely rejects, if all paths reject and all queries are $< |w|$.

For a nondeterministic Turing machine $M$ we say that the computation $M^w(x)$ is defined, if it definitely accepts or definitely rejects. For a polynomial-time oracle transducer $F$, the computation $F^w(x)$ is defined if all queries are $< |w|$.\)

\section{Oracle Construction}

The following lemma is a slightly adapted variant of a result from \cite{DG19}.

\begin{lemma}
For all $y \leq |w|$ and all $v \supseteq w$ it holds $(y \in K^v \iff y \in K^w)$.
\end{lemma}

\begin{proof}
We may assume $y = \langle 0^i, 0^j, x \rangle$ for suitable $i \in \mathbb{N}^+$ and $t, x \in \mathbb{N}$, since otherwise, $y \notin K^w$ and $y \notin K^w$. For each $q$ that is queried within the first $t$ steps of $M^w_i(x)$ or $M^w_j(x)$ it holds that $|q| \leq t < |y|$ and thus, $q < y$. Hence, these queries are answered the same way relative to $w$ and $v$, showing that $M^w_i(x)$ accepts within $t$ steps if and only if $M^w_j(x)$ accepts within $t$ steps.\)

\begin{theorem}
There exists an oracle $O$ such that the following statements hold:
\begin{itemize}
    \item DisjNP$^O$ does not contain pairs that are $\leq_{\text{poly}}^O$-hard for NP$^O \cap \text{coNP}^O$.
    \item Each $L \in \text{NP}^O$ has P$^O$-optimal proof systems.
    \item UP$^O$ does not contain $\leq_{\text{poly}}^O$-complete problems.
\end{itemize}
\end{theorem}

The following Corollary immediately follows from Theorem 3.2.

\begin{corollary}
There exists an oracle $O$ such that the following statements hold:
\begin{itemize}
    \item DisjNP$^O$ does not contain $\leq_{\text{poly}}^O$-complete pairs.
    \item Each $L \in \text{NP}^O$ has P$^O$-optimal proof systems.
    \item UP$^O$ does not contain $\leq_{\text{poly}}^O$-complete problems.
    \item NP$^O \cap \text{coNP}^O$ does not contain $\leq_{\text{poly}}^O$-complete problems.
\end{itemize}
\end{corollary}
Proof of Theorem 3.2 Let $D$ be a (possibly partial) oracle and $p \in \mathbb{P}_3$ (resp., $q \in \mathbb{P}_1$). Recall $\mathbb{P}_3 = \mathbb{P} \cap \{4k + 3 \mid k \in \mathbb{N}\}$ and $\mathbb{P}_1 = \mathbb{P} \cap \{4k + 1 \mid k \in \mathbb{N}\}$. We define

$$
A_p^D := \{0^{\nu^k} \mid k \in \mathbb{N}^+; \exists x \in \Sigma^k x \in D \text{ and } x \text{ odd}\} \cup \{0^{\nu^k} \mid k \in \mathbb{N}^+\}
$$

$$
B_p^D := \{0^{\nu^k} \mid k \in \mathbb{N}^+; \exists x \in \Sigma^k x \in D \text{ and } x \text{ even}\}
$$

$$
C_q^D := \{0^{\nu^k} \mid k \in \mathbb{N}^+; \exists x \in \Sigma^k x \in D\}
$$

Note that $A_p^D, B_p^D \in \text{NP}^D$ and $A_p^D = \overline{B_p^D}$ if $|\Sigma^k \cap D| = 1$ for each $k \in \mathbb{N}^+$. In that case $A_p^D \in \text{NP}^D \cap \text{coNP}^D$. Moreover, $C_q^D \in \text{UP}^D$ if $|\Sigma^k \cap D| \leq 1$ for each $k \in \mathbb{N}^+$.

For the sake of simplicity, let us call a pair $(M_i, M_j)$ an NP$^D \cap$coNP$^D$-machine if $L(M_i^D) = \overline{L(M_j^D)}$. Note that throughout this proof we sometimes omit the oracles in the superscript, e.g., we write NP or $A_p$ instead of NP$^D$ or $A_p^D$. However, we do not do that in the “actual” proof but only when explaining ideas in a loose way in order to give the reader the intuition behind the occasionally very technical arguments.

**Preview of construction.** We sketch some of the very basic ideas our construction uses.

1. For all positive $i \neq j$ the construction tries to achieve that $(M_i, M_j)$ is not an NP $\cap$ coNP-machine. If this is not possible, then $(L(M_i), L(M_j))$ inherently is an NP $\cap$ coNP-machine. Once we know this, we choose some odd prime $p$ and diagonalize against all FP-functions such that $A_p = \overline{B_p}$ and $A_p$ is not $\leq_m$-reducible to $(L(M_i), L(M_j))$.

2. For all $i \geq 1$ the construction intends to make sure that $F_i$ is not a proof system for $K$. If this is not possible, then $F_i$ inherently is a proof system for $K$. Then we start to encode the values of $F_i$ into the oracle. However, it is important to also allow encodings for functions that are not known to be proof systems for $K$ yet. Regarding the P-optimal proof systems, our construction is based on ideas by Dose and Gläßer [DG19].

3. For all $i \geq 1$ the construction tries to ensure that $M_i$ is not a UP-machine. In case this is impossible, we know that $M_i$ inherently is a UP-machine, which enables us to diagonalize against all FP-functions making sure that $C_q$ for some $q$ that we choose is not reducible to $L(M_i)$.

For $i \in \mathbb{N}^+$ and $x, y \in \mathbb{N}$ we write $c(i, x, y) := \langle 0^i, 0^{|x|^i + i}, 0^{|x|^i + i}, x, y, y \rangle$. Note that $|c(i, x, y)|$ is even and $|c(i, x, y)| > 4 \cdot \max(|x|^i + i, |y|)$ (cf. the properties of the pairing function $\langle \rangle$).

**Claim 3.4** Let $w \in \Sigma^*$ be an oracle, $i \in \mathbb{N}^+$, and $x, y \in \mathbb{N}$ such that $c(i, x, y) \leq |w|$. Then the following holds.

1. $F_i^w(x)$ is defined and $F_i^w(x) < |w|$.

2. $F_i^w(x) \in K^w \Leftrightarrow F_i^w(x) \in K^v$ for all $v \supseteq w$.

**Proof** As the running time of $F_i^w(x)$ is bounded by $|x|^i + i < |c(i, x, y)| < c(i, x, y) \leq |w|$, the computation $F_i^w(x)$ is defined and its output is less than $|w|$. Hence, 1 holds. Consider 2. It suffices to show that $K^v(q) = K^w(q)$ for all $q < |w|$ and all $v \supseteq w$. This holds by Lemma 3.1.

\[ \square \]

During the construction we maintain a growing collection of requirements that is represented
by a partial function belonging to the set

\[ T = \left\{ t : \mathbb{N}^+ \cup (\mathbb{N}^+)^2 \rightarrow \mathbb{Z} \mid \text{dom}(t) \text{ is finite}, \, t \text{ is injective on its support}, \right. \]

\[ \begin{align*}
& \quad \left. \bullet \ t(\mathbb{N}^+) \subseteq \{0\} \cup \mathbb{N}^+ \\
& \quad \bullet \ t(\{(i, i) \mid i \in \mathbb{N}^+\}) \subseteq \{0\} \cup \{ -q \mid q \in \mathbb{P}_1 \} \\
& \quad \bullet \ t(\{(i,j) \in (\mathbb{N}^+)^2 \mid i \neq j\}) \subseteq \{0\} \cup \{ -p \mid p \in \mathbb{P}_3 \} \right\}. \]

A partial oracle \( w \in \Sigma^* \) is called \( t \)-valid for \( t \in T \) if it satisfies the following properties.

**V1** For all \( i \in \mathbb{N}^+ \) and all \( x, y \in \mathbb{N} \), if \( c(i, x, y) \in w \), then \( F_i^w(x) = y \in K^w \).

(meaning: if the oracle contains the codeword \( c(i, x, y) \), then \( F_i^w(x) \) outputs \( y \) and \( y \in K^w \); hence, \( c(i, x, y) \in w \) is a proof for \( y \in K^w \)).

**V2** For all distinct \( i, j \in \mathbb{N}^+ \), if \( t(i, j) = 0 \), then there exists \( x \) such that \( M_i^w(x) \) and \( M_j^w(x) \) definitely accept.
(meaning: for every extension of the oracle, \( (L(M_i), L(M_j)) \) is not a disjoint NP-pair.)

**V3** For all distinct \( i, j \in \mathbb{N}^+ \) with \( t(i, j) = -p \) for some \( p \in \mathbb{P}_3 \) and each \( k \in \mathbb{N}^+ \), it holds (i) \( |\Sigma^k \cap w| \leq 1 \) and (ii) if \( w \) is defined for all words of length \( p^k \), then \( |\Sigma^k \cap w| = 1 \).

(meaning: if \( t(i, j) = -p \), then ensure that \( A_p = \overline{B_p} \) (i.e., \( A_p \in \text{NP} \cap \text{coNP} \)) relative to the final oracle.)

**V4** For all \( i \in \mathbb{N}^+ \) with \( t(i) = 0 \), there exists \( x \) such that \( F_i^w(x) \) is defined and \( F_i^w(x) \notin K^w \) for all \( v \supseteq w \).

(meaning: for every extension of the oracle, \( F_i \) is not a proof system for \( K \))

**V5** For all \( i \in \mathbb{N}^+ \) and \( x \in \mathbb{N} \) with \( 0 < t(i) \leq c(i, x, F_i^w(x)) < |w| \), it holds \( c(i, x, F_i^w(x)) \in w \).
(meaning: if \( t(i) > 0 \), then from \( t(i) \) on, we encode the values of \( F_i \) into the oracle.
Note that V5 is not in contradiction with e.g. V3 or V7 as \(|c(\cdot, \cdot, \cdot)|\) is even.)

**V6** For all \( i \in \mathbb{N}^+ \) with \( t(i, i) = 0 \), there exists \( x \) such that \( M_i^w(x) \) is defined and has two accepting paths.
(meaning: for every extension of the oracle, \( M_i \) is not a UP-machine.)

**V7** For all \( i \in \mathbb{N}^+ \) with \( t(i, i) = -q \in \mathbb{P}_1 \) and each \( k \in \mathbb{N}^+ \), it holds \( |\Sigma^k \cap w| \leq 1 \).

(meaning: if \( t(i, i) = -q \), ensure that \( C_q \) is in UP.)

The subsequent claim follows directly from the definition of \( t \)-valid.

**Claim 3.5** Let \( t, t' \in T \) such that \( t' \) is an extension of \( t \). For all oracles \( w \in \Sigma^* \), if \( w \) is \( t \)-valid, then \( w \) is \( t' \)-valid.

**Claim 3.6** Let \( t \in T \) and \( u, v, w \in \Sigma^* \) be oracles such that \( u \subseteq v \subseteq w \) and both \( u \) and \( w \) are \( t \)-valid. Then \( v \) is \( t \)-valid.

**Proof** \( v \) satisfies V2, V4, and V6 since \( u \) satisfies these conditions. Moreover, \( v \) satisfies V3 and V7 as \( w \) satisfies these conditions.

Let \( i \in \mathbb{N}^+ \) and \( x, y \in \mathbb{N} \) such that \( c(i, x, y) \in v \). Then \( c(i, x, y) \in w \) and as \( w \) is \( t \)-valid, we obtain by V1 that \( F_i^w(x) = y \in K^w \). Claim 3.4 yields that \( F_i^w(x) \) is defined and \( F_i^w(x) \in K^w \Leftrightarrow F_i^w(x) \in K^w \). This yields that \( F_i^w(x) = F_i^w(x) = y \) and \( K^w(y) = K^w(y) = 1 \). Thus, \( v \) satisfies V1.
Now let \( i \in \mathbb{N}^+ \) and \( x \in \mathbb{N} \) such that \( 0 < t(i) \leq c(i, x, F^i_v(x)) < |v| \). Again, by Claim 3.4, \( F^i_v(x) \) is defined and thus, \( F^i_v(x) = F^i_{v'}(x) \). As \( |v| \leq |w| \) and \( w \) is \( t \)-valid, we obtain by V5 that \( c(i, x, F^i_v(x)) = c(i, x, F^i_{v'}(x)) \in w \). Since \( v \sqsubseteq w \) and \( |v| > c(i, x, F^i_v(x)) \), we obtain \( c(i, x, F^i_v(x)) \in v \), which shows that \( v \) satisfies V5.

**Oracle construction.** Let \( T \) be an enumeration of \( \bigcup_{i=1}^3 (\mathbb{N}^+)^i \) having the property that \((i, j)\) appears earlier than \((i', j, r)\) for all \( i, j, r \in \mathbb{N}^+ \) (more formally, \( T \) could be defined as a function \( \mathbb{N} \to \bigcup_{i=1}^3 (\mathbb{N}^+)^i \)). Each element of \( T \) stands for a task. We treat the tasks in the order specified by \( T \) and after treating a task we remove it and possibly other tasks from \( T \). We start with the nowhere defined function \( t_0 \) and the \( t_0 \)-valid oracle \( w_0 = \varepsilon \). Then we define functions \( t_1, t_2, \ldots \) in \( T \) such that \( t_{i+1} \) is an extension of \( t_i \) and partial oracles \( w_0 \sqsupseteq w_1 \sqsupseteq w_2 \sqsupseteq \ldots \) such that each \( w_i \) is \( t_i \)-valid. Finally, we choose \( O = \bigcup_{i=0}^\infty w_i \) (note that \( O \) is totally defined since in each step we will strictly extend the oracle). We describe step \( s > 0 \), which starts with some \( t_{s-1} \in T \) and a \( t_{s-1} \)-valid oracle \( w_{s-1} \) and chooses an extension \( t_s \in T \) of \( t_{s-1} \) and a \( t_s \)-valid \( w_s \sqsupseteq w_{s-1} \) (it will be argued later that all these steps are indeed possible). Let us recall that each task is immediately deleted from \( T \) after it is treated.

- **task \((i, i)\):** Let \( t' = t_{s-1} \cup \{i \mapsto 0\} \). If there exists a \( t' \)-valid \( v \sqsupseteq w_{s-1} \), then let \( t_s = t' \) and \( w_s \) be the least \( t' \)-valid, partial oracle \( w_{s-1} \). Otherwise, let \( t_s = t_{s-1} \cup \{i \mapsto |w_{s-1}|\} \) and choose \( w_s = w_{s-1}b \) with \( b \in \{0, 1\} \) such that \( w_s \) is \( t_s \)-valid.
  (meaning: try to ensure that \( F_i \) is not a proof system for \( K \). If this is impossible, require that from now on \( F_i \) are encoded into the oracle.)

- **task \((i, j)\) with \( i \neq j\):** Let \( t' = t_{s-1} \cup \{(i, j) \mapsto 0\} \). If there exists a \( t' \)-valid \( v \sqsupseteq w_{s-1} \), then let \( t_s = t' \) and define \( w_s \) to be the least \( t' \)-valid, partial oracle \( w_{s-1} \), and delete all tasks \((i, j, \cdot)\) from \( T \). Otherwise, let \( z = |w_{s-1}| \), choose some \( p \in \mathbb{F}_3 \) greater than \( |z| \) and all \( p' \) with \( p' \in \mathbb{P}_{\geq 3} \) and \( \neg p' \in \text{ran}(t_{s-1}) \), let \( t_s = t_{s-1} \cup \{(i, j) \mapsto \neg p\} \), and choose \( w_s = w_{s-1}b \) with \( b \in \{0, 1\} \) such that \( w_s \) is \( t_s \)-valid.
  (meaning: try to ensure that \( (L(M_i), L(M_j)) \) is not a disjoint NP-pair. If this is impossible, choose a sufficiently large prime \( p \). It will be made sure later that \( A_p \) cannot be reduced to \( (L(M_i), L(M_j)) \).)

- **task \((i, j, r)\) with \( i \neq j\):** It holds \( t_{s-1}(i, j) = \neg p \) for a prime \( p \in \mathbb{F}_3 \), since otherwise, this task would have been deleted in the treatment of task \((i, j)\). Define \( t_s = t_{s-1} \) and choose a \( t_s \)-valid \( w_s \sqsupseteq w_{s-1} \) such that for some \( n \in \mathbb{N}^+ \) one of the following two statements holds:
  - \( 0^n \in A^p_w \) for all \( v \sqsupseteq w_s \) and \( M_i^{w_s}(F^{w_s}_r(0^n)) \) definitely rejects.
  - \( 0^n \in B^p_w \) for all \( v \sqsupseteq w_s \) and \( M_j^{w_s}(F^{w_s}_r(0^n)) \) definitely rejects.
  (meaning: make sure that it does not hold \( (A_p, B_p) \equiv \text{pp}(L(M_i), L(M_j)) \) via \( F_r \). Due to V3 it will hold \( A_p = \text{B}^p_p \) relative to the final oracle and hence, it will not hold \( A_p \equiv \text{pp}(L(M_i), L(M_j)) \) via \( F_r \).)

- **task \((i, i)\):** Let \( t' = t_{s-1} \cup \{(i, i) \mapsto 0\} \). If there exists a \( t' \)-valid \( v \sqsupseteq w_{s-1} \), then let \( t_s = t' \), define \( w_s \) to be the least \( t' \)-valid, partial oracle \( w_{s-1} \), and delete all tasks \((i, i, \cdot)\) from \( T \). Otherwise, let \( z = |w_{s-1}| \), choose some \( q \in \mathbb{F}_1 \) greater than both \( |z| \) and all \( p' \) with \( p' \in \mathbb{P}_{\geq 3} \) and \( \neg p' \in \text{ran}(t_{s-1}) \), let \( t_s = t_{s-1} \cup \{(i, i) \mapsto \neg q\} \), and choose \( w_s = w_{s-1}b \) with \( b \in \{0, 1\} \) such that \( w_s \) is \( t_s \)-valid.
  (meaning: try to ensure that \( M_i \) is not a UP-machine. If this is impossible, choose a sufficiently large prime \( q \in \mathbb{F}_1 \). It will be made sure later that \( C_q \) cannot be reduced to \( L(M_i) \).)
• task \((i, i, r)\): It holds \(t_{s-1}(i, j) = -q\) for a prime \(q \in \mathbb{P}_1\), since otherwise, this task would have been deleted in the treatment of task \((i, i)\). Define \(t_s = t_{s-1}\) and choose a \(t_s\)-valid \(w_s \supseteq w_{s-1}\) such that for some \(n \in \mathbb{N}^+\) one of the following conditions holds:

- \(0^n \in C_q^w\) for all \(v \supseteq w_s\) and \(M_{i}^{w_s}(F_r^{w_s}(0^n))\) definitely rejects.
- \(0^n \notin C_q^w\) for all \(v \supseteq w_s\) and \(M_{i}^{w_s}(F_r^{w_s}(0^n))\) definitely accepts.

(meaning: make sure that it does not hold \(C_q^{w_s} \subseteq \mathbb{N}_0 L(M_i)\) via \(F_r\)).

Observe that \(t_s\) is always chosen in a way such that it is in \(T\). We now show that the construction is possible. For that purpose, we first describe how a valid oracle can be extended by one bit such that it remains valid.

**Claim 3.7** Let \(s \in \mathbb{N}\) and \(w \in \Sigma^*\) be a \(t_s\)-valid oracle with \(w \supseteq w_s\). It holds for \(z = |w|\):

1. If \(z = c(i, x, y)\) for \(i \in \mathbb{N}^+\) and \(x, y \in \mathbb{N}\), \(0 < t_s(i) < z\), and \(F_i^{w}(x) = y\), then \(F_i^{w}(x)\) is defined and \(y \in K^w\) for all \(v \supseteq w\).

2. There exists \(b \in \{0, 1\}\) such that \(w b\) is \(t_s\)-valid. In detail, the following statements hold.

   (a) If \(|z|\) is odd and for all \(p \in \mathbb{P}\) and \(k \in \mathbb{N}^+\) with \(-p \in \text{ran}(t_s)\) it holds \(|z| \neq p^k\), then \(w 0\) and \(w 1\) are \(t_s\)-valid.

   (b) If there exist \(p \in \mathbb{P}_3\) and \(k \in \mathbb{N}^+\) with \(-p \in \text{ran}(t_s)\) such that \(|z| = p^k\), \(z \neq 1^p^k\), and \(w \cap \Sigma^p^k = \emptyset\), then \(w 0\) and \(w 1\) are \(t_s\)-valid.

   (c) If there exist \(p \in \mathbb{P}_3\) and \(k \in \mathbb{N}^+\) with \(-p \in \text{ran}(t_s)\) such that \(z = 1^p^k\) and \(w \cap \Sigma^p^k = \emptyset\), then \(w 0\) and \(w 1\) are \(t_s\)-valid.

   (d) If there exist \(q \in \mathbb{P}_1\) and \(k \in \mathbb{N}^+\) with \(-q \in \text{ran}(t_s)\) such that \(|z| = q^k\) and \(w \cap \Sigma^q^k = \emptyset\), then \(w 0\) and \(w 1\) are \(t_s\)-valid.

   (e) If \(z = c(i, x, y)\) for \(i \in \mathbb{N}^+\) and \(x, y \in \mathbb{N}\), \(0 < t_s(i) \leq z\), and \(F_i^{w}(x) = y\), then \(w 1\) is \(t_s\)-valid and \(F_i^{w 1}(x) = y\).

   (f) If \(z = c(i, x, y)\) for \(i \in \mathbb{N}^+\) and \(x, y \in \mathbb{N}\), at least one of the three conditions (i) \(t_s(i)\) undefined, (ii) \(t_s(i) = 0\), and (iii) \(t_s(i) > z\) holds, and \(F_i^{w}(x) = y \in K^w\), then \(w 0\) and \(w 1\) are \(t_s\)-valid.

   (g) In all other cases (i.e., none of the assumptions in (2a)-(2f) holds) \(w 0\) is \(t_s\)-valid.

**Proof**

1. By Claim 3.4, \(F_i^{w}(x)\) is defined. Assume that for \(z = |w|\) it holds \(z = c(i, x, y)\) for \(i \in \mathbb{N}^+\) and \(x, y \in \mathbb{N}\), \(0 < t_s(i) \leq z\), and \(F_i^{w}(x) = y \notin K^w\). Let \(s' > 0\) be the step where the task \(i\) is treated (note \(s' < s\) as \(t_s(i)\) is defined). By Claim 3.5, \(w\) is \(t_{s' - 1}\)-valid. Moreover, by Claim 3.4, \(F_i^{w}(x) \notin K^w\) for all \(v \supseteq w\). Thus, \(w\) is \(t'\)-valid for \(t' = t_{s' - 1} \cup \{i \mapsto 0\}\), which is why the construction would have chosen \(t_{s'} = t'\), in contradiction to \(t_s(i) > 0\). Hence, \(y \in K^w\) and by Claim 3.4, it even holds \(y \in K^w\) for all \(v \supseteq w\). This shows statement 1.

2. We first show the following assertions.

   \(w 0\) satisfies V1. \hspace{1cm} (1)

   If (i) \(z = c(i, x, y)\) for \(i \in \mathbb{N}^+\) and \(x, y \in \mathbb{N}\) with \(F_i^{w}(x) = y \in K^w\) or (ii) \(z\) has odd length, then \(w 1\) satisfies V1. \hspace{1cm} (2)

   \(w 0\) satisfies V5 unless there exist \(i \in \mathbb{N}^+\) and \(x, y \in \mathbb{N}\) such that (i) \(z = c(i, x, y)\), (ii) \(0 < t_s(i)\), (iii) \(t_s(i) \leq z\), and (iv) \(F_i^{w}(x) = y\) \hspace{1cm} (3)

   \(w 1\) satisfies V5. \hspace{1cm} (4)
(1) and (2): Let \( i' \in \mathbb{N}^+ \) and \( x', y' \in \mathbb{N} \) such that \( c(i', x', y') \in w \). Then, as \( w \) is \( t_s \)-valid, by V1, \( F_i^w(x') = y' \in K^w \) and by Claim 3.4, \( F_i^w(x') \) is defined and \( y' \in K^w \) for all \( v \supseteq w \). Hence, in particular, \( F_i^{w_b}(x') = y' \in K^{w_b} \) for all \( b \in \{0, 1\} \). This shows (1). For the proof of (2) it remains to consider \( z \). In case (ii) \( w \) satisfies V1 as \( |z| \) is odd and each \( c(i, x, y) \) has even length. Consider case (i), i.e., \( z = c(i, x, y) \) for \( i \in \mathbb{N}^+ \) and \( x, y \in \mathbb{N} \) with \( F_i^w(x) = y \in K^w \). Then by Claim 3.4, \( F_i^{w_1}(x) = y \in K^{w_1} \), which shows (2).

(3) and (4): Let \( i' \in \mathbb{N}^+ \) and \( x' \in \mathbb{N} \) such that \( 0 < t_s(i') \leq c(i', x', F_i^w(x')) < |w| \). Then by Claim 3.4, \( F_i^w(x') \) is defined and thus, \( F_i^{w_b}(x') = F_i^w(x') \) for all \( b \in \{0, 1\} \). As \( w \) is \( t_s \)-valid, it holds \( c(i', x', F_i^w(x')) \in w \) and hence, \( c(i', x', F_i^{w_b}(x')) \in w \subseteq w_b \) for all \( b \in \{0, 1\} \). This shows (4). In order to finish the proof for (3), it remains to consider \( z \). Assume \( z = c(i, x, y) \) for some \( i, x, y \in \mathbb{N} \) with \( i > 0 \) (otherwise, \( w_0 \) clearly satisfies V5). If (ii) or (iii) is wrong, then \( w_0 \) satisfies V5. If (iv) is wrong, then \( F_i^{w_0}(x) \neq y \). By Claim 3.4, this computation is defined and hence, \( F_i^{w_0}(x) \neq y \), which is why \( w_0 \) satisfies V5. This shows (3).

Let us now prove the assertions (2a)-(2g) and note that we do not have to consider V2, V4, and V6 as these conditions are not affected by extending a \( t_s \)-valid oracle.

(a) By (1) and (2), the oracles \( w_0 \) and \( w_1 \) satisfy V1. By (3) and (4), the oracles \( w_0 \) and \( w_1 \) satisfy V5 (for the application of (3) recall that each \( c(i, x, y) \) has even length and hence, for all \( i, x, y \) condition (i) does not hold). V3 and V7 are not affected as \( |z| \neq p^k \) for all primes \( p \) with \(-p \in \text{ran}(t_s) \) and all \( k > 0 \).

(b) By (1), (2), (3), and (4), the oracles \( w_0 \) and \( w_1 \) satisfy V1 and V5 (for the application of (3) recall that each \( c(i, x, y) \) has even length and hence, for all \( i, x, y \) condition (i) does not hold). As \( p \in \mathbb{P}_3 \), V7 is satisfied by \( w_0 \) and \( w_1 \). Moreover, \( w_0 \) satisfies V3 as due to \( z \neq 1p^k \) the oracle \( w_0 \) is not defined for all words of length \( p^k \). Finally, \( w_1 \) satisfies V3 since \( \Sigma^p \cap w = \emptyset \).

(c) By (2) and (4), the oracle \( w_1 \) satisfies V1 and V5. As \( p \in \mathbb{P}_3 \), V7 is satisfied by \( w_1 \). Moreover, as \( w \cap \Sigma^p = \emptyset \), it holds \( |w_1 \cap \Sigma^p| = 1 \) and hence, \( w_1 \) satisfies V3.

(d) By (1), (2), (3), and (4), the oracles \( w_0 \) and \( w_1 \) satisfy V1 and V5 (for the application of (3) recall that each \( c(i, x, y) \) has even length and hence, for all \( i, x, y \) condition (i) does not hold). As \( q \in \mathbb{P}_1 \), the oracles \( w_0 \) and \( w_1 \) satisfy V3. Finally, \( w_0 \) trivially satisfies V7 and \( w_1 \) satisfies V7 as \( w \cap \Sigma^q = \emptyset \).

(e) By (4), the oracle \( w_1 \) satisfies V5. By statement 1 of the current claim, \( F_i^w(x) \) is defined and \( y \in K^v \) for all \( v \supseteq w \). Hence, (2) can be applied, \( w_1 \) satisfies V1, and \( F_i^{w_1}(x) = F_i^w(x) = y \). As \( |z| \) is even, \( w_1 \) trivially satisfies V3 and V7.

(f) By (1), \( w_0 \) satisfies V1. By (2), \( w_1 \) satisfies V1. By (3), \( w_0 \) satisfies V5. By (4), \( w_1 \) satisfies V5. As \( |z| \) is even, both \( w_0 \) and \( w_1 \) satisfy V3 and V7.

(g) By (1), \( w_0 \) satisfies V1. Moreover, (3) can be applied since otherwise, there would exist \( i, x, y \in \mathbb{N} \) with \( i > 0 \) such that conditions (i)-(iv) of the assertion (3) hold and then we were in case 2(e). Hence, \( w_0 \) satisfies V5. Trivially, \( w_0 \) satisfies V7 and finally, \( w_0 \) satisfies V3 as the only way \( w_0 \) could hurt V3 is that \( z = 1p^k \) for some \( p \in \mathbb{P}_3 \) with \(-p \in \text{ran}(t_s) \) and \( k > 0 \) as well as \( w \cap \Sigma^p = \emptyset \), but this case is treated in 2(c).

This finishes the proof of Claim 3.7. \( \square \)

In order to show that the above construction is possible, assume that it is not possible and let \( s > 0 \) be the least number, where it fails.

If step \( s \) treats a task \( t \in \mathbb{N}^+ \cup \mathbb{N}^+ \), then \( t_{s-1}(t) \) is not defined, since the value of \( t \) is defined in the unique treatment of the task \( t \). If \( t_s(t) \) is chosen to be 0, then the construction clearly is possible. Otherwise, due to the (sufficiently large) choice of \( t_s(t) \), the \( t_{s-1} \)-valid oracle \( w_{s-1} \)
is even $t_s$-valid and Claim 3.7.2 ensures that there exists a $t_s$-valid $w_{s-1}b$ for some $b \in \{0, 1\}$. Hence, the construction does not fail in step $s$, a contradiction.

For the remainder of the proof that the construction above is possible we assume that step $s$ treats a task $(i, j, r) \in (\mathbb{N}^+)^3$. We treat the cases $i = j$ and $i \neq j$ simultaneously whenever it is possible. Recall that in the case $i = j$ we work for the diagonalization ensuring that $L(M_1)$ is not a complete UP-set and in the case $i \neq j$ we work for the diagonalization ensuring that the pair $(L(M_i), L(M_j))$ is not hard for NP $\cap$ coNP.

In both cases, $t_s = t_{s-1}$ and $t_s(i, j) = -p$ for some $p \in \mathbb{P}^3$ (recall $p \in \mathbb{P}_1$ if $i = j$ and $p \in \mathbb{P}_3$ if $i \neq j$). Let $\gamma(x) = (x'+r)^{i+j} + i+j$ and choose $n = p^k$ for some $k \in \mathbb{N}^+$ such that

$$2^{2n-2} > 2^{n+1} \cdot \gamma(n) \tag{5}$$

and $w_{s-1}$ is not defined for any words of length $n$. Note that $\gamma(n)$ is not less than the running time of each of the computations $M_i^D(F_{r}^D(0^n))$ and $M_j^D(F_{r}^D(0^n))$ for each oracle $D$.

We define $u \supseteq w_{s-1}$ to be the minimal $t_s$-valid oracle that is defined for all words of length $< n$. Such an oracle exists by Claim 3.7.2.

Moreover, for $z \in \Sigma^n$, let $u_z \supseteq u$ be the minimal $t_s$-valid oracle with $u_z \cap \Sigma^n = \{z\}$ that is defined for all words of length $\leq \gamma(n)$. Such an oracle exists by Claim 3.7.2: first, starting from $u$ we extend the current oracle bitwise such that (i) it remains $t_s$-valid, (ii) it is defined for precisely the words of length $\leq n$, and (iii) its intersection with $\Sigma^n$ equals $\{z\}$. This is possible by (2b, 2c, and 2g) or (2d and 2g) of Claim 3.7 depending on whether $p \in \mathbb{P}_3$ or $p \in \mathbb{P}_1$. Then by Claim 3.7.2, the current oracle can be extended bitwise without losing its $t_s$-validity until it is defined for all words of length $\leq \gamma(n)$.

Claim 3.8 Let $z \in \Sigma^n$.

1. For each $\alpha \in u_z \cap \Sigma^{>n}$ one of the following statements holds.
   - $\alpha = 1^{p^\alpha}$ for some $p' \in \mathbb{P}_3$ with $-p' \in \text{ran}(t_s)$ and some $\kappa > 0$.
   - $\alpha = c(i', x, y)$ for some $i' \in \mathbb{N}^+$ and $x, y \in \mathbb{N}$ with $0 < t_s(i') \leq c(i', x, y)$, $F^u_{i'}(x) = y$, and $y \in \Sigma^u_z$.

2. For all $p' \in \mathbb{P}_3$ with $-p' \in \text{ran}(t_s)$ and all $\kappa > 0$, if $n < p^{\kappa} \leq \gamma(n)$, then $u_z \cap \Sigma^{p^{\kappa}} = \{1^{p^\alpha}\}$.

Proof

1. Let $\alpha \in u_z \cap \Sigma^{>n}$. Moreover, let $u'$ be the prefix of $u_z$ that has length $\alpha$, i.e., $\alpha$ is the least word that $u'$ does not define for. In particular, it holds $u' \cap \Sigma^{\leq n} = u_z \cap \Sigma^{\leq n}$ and thus, $u' \cap \Sigma^n = \{z\}$. As $u \subseteq u' \subseteq u_z$ and both $u$ and $u_z$ are $t_s$-valid, Claim 3.6 yields that $u'$ is also $t_s$-valid.

Let us apply Claim 3.7.2 to the oracle $u'$. If one of the cases 2a, 2b, 2d, 2f, and 2g can be applied, then $u'0$ is $t_s$-valid and can be extended to a $t_s$-valid oracle $u''$ with $|u''| = |u_z|$ by Claim 3.7.2. As $u''$ and $u_z$ agree on all words $< \alpha$ and $\alpha \in u_z - u''$, we obtain $u'' < u_z$ and due to $u' \subseteq u''$ we know that $u'' \cap \Sigma^n = \{z\}$. This is a contradiction to the choice of $u_z$ (recall that $u_z$ is the minimal $t_s$-valid oracle that is defined for all words of length $\leq \gamma(n)$ and that satisfies $u_z \cap \Sigma^n = \{z\}$).

Hence, none of the cases 2a, 2b, 2d, 2f, and 2g of Claim 3.7 can be applied, i.e., either (i) $\alpha = 1^{p^\alpha}$ for some $p' \in \mathbb{P}_3$ and $\kappa > 0$ with $-p' \in \text{ran}(t_s)$ or (ii) $\alpha = c(i', x, y)$ for $i', x, y \in \mathbb{N}$, $i' > 0$, $0 < t_s(i') \leq \alpha$, and $F^u_{i'}(x) = y$. In the latter case Claim 3.7.1 shows that $F^u_{i'}(x)$ is defined and $y \in \Sigma^n$ for all $v \supseteq u'$, which implies $F^u_{i'}(x) = y \in \Sigma^u_z$. 

12
2. As \( -p' \in \text{ran}(t_s) \), \( u_z \) is \( t_s \)-valid, and \( u_z \) is defined for all words of length \( p^n \), \( V3 \) yields that there exists \( \beta \in \Sigma^{p^n} \cap u_z \). Let \( \beta \) be the minimal element of \( \Sigma^{p^n} \cap u_z \). It suffices to show \( \beta = 1^{p^n} \). For a contradiction, we assume \( \beta < 1^{p^n} \). Let \( u' \) be the prefix of \( u_z \) that is defined for exactly the words of length \( < p^n \). Then \( u \sqsubseteq u' \sqsubseteq u_z \) and both \( u \) and \( u_z \) are \( t_s \)-valid. Hence, by Claim 3.6, the oracle \( u' \) is \( t_s \)-valid as well.

By Claim 3.7.2, \( u' \) can be extended to a \( t_s \)-valid oracle \( u'' \) that satisfies \( |u''| = |u_z| \) and \( u'' \cap \Sigma^{p^n} = \{1^{p^n}\} \). Then \( \beta \in u_z - u'' \). As the oracles \( u'' \) and \( u_z \) agree on all words smaller than \( \beta \), we have \( u'' < u_z \) and \( u'' \cap \Sigma^n = \{z\} \), in contradiction to the choice of \( u_z \) (again, recall that \( u_z \) is the minimal \( t_s \)-valid oracle that is defined for all words of length \( \leq \gamma(n) \) and that satisfies \( u_z \cap \Sigma^n = \{z\} \)).

This finishes the proof of Claim 3.8. \( \square \)

Let us study the case that for some odd (resp., even) \( z \in \Sigma^n \) the computation \( M^u_z(F^{u_z}_r(0^n)) \) (resp., \( M^u_z(F^{u_z}_r(0^n)) \)) if \( z \) is even) rejects. Then it even definitely rejects since \( u_z \) is defined for all words of length \( \gamma(n) \). If \( i \neq j \), then \( p \in P_3 \) and since \( z \in u_z \), we have \( 0^n \in A^u_p \) for all \( v \sqsupseteq u_z \) (resp., \( 0^n \in B^u_p \) for all \( v \sqsupseteq u_z \) if \( z \) is even). Analogously, if \( i = j \), then \( p \in P_1 \) and as \( z \in u_z \), we have \( 0^n \in C^u_p \) for all \( v \sqsupseteq u_z \). Hence, in all these cases we can choose \( w_s = u_z \) and obtain a contradiction to the assumption that step \( s \) of the construction fails in treating the task \( (i,j,r) \). Therefore, for the remainder of the proof that the construction is possible we assume the following:

- For each odd \( z \in \Sigma^n \) the computation \( M^u_z(F^{u_z}_r(0^n)) \) definitely accepts.
- For each even \( z \in \Sigma^n \) the computation \( M^u_z(F^{u_z}_r(0^n)) \) definitely accepts.

Note that in case \( i = j \) we could have also formulated the two conditions equivalently in the following simpler way: for each \( z \in \Sigma^n \) the computation \( M^u_z(F^{u_z}_r(0^n)) \) definitely accepts.

Recall, however, that as far as possible we consider the cases \( i = j \) and \( i \neq j \) simultaneously.

Let \( U_z \) for \( z \in \Sigma^n \) odd (resp., \( z \in \Sigma^n \) even) be the set of all those oracle queries of the least accepting path of \( M^u_z(F^{u_z}_r(0^n)) \) (resp., \( M^u_z(F^{u_z}_r(0^n)) \)) that are of length \( \geq n \). Observe \( \ell(U_z) \leq \gamma(n) \). Moreover, define \( Q_0(U_z) = U_z \) and for \( m \in \mathbb{N} \),

\[
Q_{m+1}(U_z) = \bigcup_{c(i',x,y) \in Q_m(U_z)} \left\{ q \in \Sigma^{\geq n} \mid q \text{ is queried by } F^{u_z}_{i'}(x) \right\} \cup \left\{ q \in \Sigma^{\geq n} \mid y = (0^{n_1},0^{n_2})^{i''+1^{n_2}}, x' \text{ for some } i'' > 0 \text{ and } x' \in \mathbb{N}, M^{u_z}_{i''}(x') \text{ has an accepting path and } q \text{ is queried by the least such path } \right\}.
\]

Let \( Q(U_z) = \bigcup_{m \in \mathbb{N}} Q_m(U_z) \). Note that all words in \( Q(U_z) \) have length \( \geq n \). Moreover, note that for \( c(i',x,y) \in Q_m(U_z) \) for some \( m \) it does not necessarily hold \( y \in K^{u_z} \) and therefore, it might be that the computation \( M^{u_z}_{i''}(x') \) (in the notation used in the equation above) does not have any accepting paths. In that case the second of the two sets in the equation above is empty.

**Claim 3.9** For all \( z \in \Sigma^n \), \( \ell(Q(U_z)) \leq 2\ell(U_z) \leq 2\gamma(n) \) and the length of each word in \( Q(U_z) \) is \( \leq \gamma(n) \).

**Proof** We show that for all \( m \in \mathbb{N} \), \( \ell(Q_m(U_z)) \leq 1/2 \cdot \ell(Q_m(U_z)) \). Then \( \sum_{s=0}^{\ell} 1/2^m \leq 2 \) for all \( s \in \mathbb{N} \) implies \( \ell(Q(U_z)) \leq 2 \cdot \ell(U_z) \). Moreover, from \( \ell(U_z) \leq \gamma(n) \) and \( \ell(Q_m(U_z)) \leq 1/2 \cdot \ell(Q_m(U_z)) \) the second part of the claim follows.
Let $m \in \mathbb{N}$ and consider an arbitrary element $\alpha$ of $Q_m(U)$. If $\alpha$ is not of the form $c(i', x, y)$ for $i' \in \mathbb{N}^+$ and $x, y \in \mathbb{N}$, then $\alpha$ generates no elements in $Q_{m+1}(U)$. Assume $\alpha = c(i', x, y)$ for $i' \in \mathbb{N}^+$ and $x, y \in \mathbb{N}$ with $y = (0,0) \oplus i'' + i'''$ for $i'' \in \mathbb{N}^+$ and $x' \in \mathbb{N}$. The computation $F_{i''}^{u_{x'}}(x)$ runs for at most $|x'| + i' < |\alpha|/4$ steps, where “$<$” holds by the definition of $c(\cdot, \cdot, \cdot)$ and the properties of the pairing function $(\cdot, \cdot)$. Hence, the set of queries $Q$ of $F_{i''}^{u_{x'}}(x)$ satisfies $\ell(Q) \leq |\alpha|/4$.

Moreover, the computation $M_{i''}^{u_{x'}}(x)$ runs for less than $|y| < |\alpha|/4$ steps (for “$<$” we refer again to the definition of $c(\cdot, \cdot, \cdot)$ and the properties of the pairing function $(\cdot, \cdot)$). Hence, for the set $Q$ of queries of the least accepting path of the computation $M_{i''}^{u_{x'}}(x)$ (if such a path exists) we have $\ell(Q) \leq |\alpha|/4$.

Consequently,

$$
\ell(Q_{m+1}(U)) \leq \sum_{c(i', x, y) \in Q_m(U)} \left[ \ell\left(\{q \in \Sigma^{\geq n} \mid q \text{ is queried by } F_{i''}^{u_{x'}}(x)\}\right)_{\leq |c(i', x, y)|/4} \right. \leq \sum_{c(i', x, y) \in Q_m(U)} \left. \ell\left(\{q \in \Sigma^{\geq n} \mid y = (0,0) \oplus i'' + i''' \text{ for some } i'' > 0 \text{ and } x' \in \mathbb{N}, M_{i''}^{u_{x'}}(x') \text{ has an accepting path and } q \text{ is queried by the least such path}\}\right) \right].
$$

which finishes the proof of Claim 3.9.

For $z, z' \in \Sigma^n$ we say that $Q(U_z)$ and $Q(U_{z'})$ conflict if there is a word $\alpha \in Q(U_z) \cap Q(U_{z'})$ which is in $u_z \cup u_{z'}$. In that case, we say $Q(U_z)$ and $Q(U_{z'})$ conflict in $\alpha$. Note that whenever $Q(U_z)$ and $Q(U_{z'})$ conflict in a word $\alpha$, then $\alpha \in u_z \cup u_{z'}$ and $|\alpha| \geq n$.

The next five claims are dedicated to the purpose of proving that for each odd $z \in \Sigma^n$ and each even $z' \in \Sigma^n$, the sets $Q(U_z)$ and $Q(U_{z'})$ conflict in a word of length $n$. Indeed, then $Q(U_z)$ and $Q(U_{z'})$ conflict in one of the words $z$ and $z'$ as these are the only words of length $n$ in $u_z \cup u_{z'}$.

Claim 3.10 Let $z, z' \in \Sigma^n$ such that $z$ is odd and $z'$ is even. If $Q(U_z)$ and $Q(U_{z'})$ conflict, then they conflict in a word of length $n$.

Proof Let $\alpha$ be the least word in which $Q(U_z)$ and $Q(U_{z'})$ conflict (note that $|\alpha| \leq \gamma(n)$ due to $\alpha \in Q(U_z) \cap Q(U_{z'})$ and Claim 3.9). Then $\alpha \in u_z \cup u_{z'}$. By symmetry, it suffices to consider the case $\alpha \in u_z - u_{z'}$. For a contradiction, assume that $|\alpha| > n$. Then by Claim 3.8, two situations are possible.

1. Assume $\alpha = 1^p \in \mathbb{P}_3$ with $-p' \in \text{ran}(t_z)$ and $\kappa > 0$. Then by Claim 3.8.2, $\alpha \in u_{z'}$, a contradiction. Hence, $\alpha \neq 1^p \in \mathbb{P}_3$ for all $p' \in \mathbb{P}_3$ with $-p' \in \text{ran}(t_z)$ and $\kappa > 0$.

2. Here, $\alpha = c(i', x, y)$ for $i' \in \mathbb{N}^+$ and $x, y \in \mathbb{N}$ with $0 < t_z(i') \leq c(i', x, y)$ and $F_{i'}^{u_{x'}}(x) = y \in K_{u_z}$. By construction, $t_z(i') = t_{z'-1}(i') \leq |u_{z'-1}| \leq |u| < |\alpha|$. Thus, $F_{i'}^{u_{x'}}(x) \neq y$, since otherwise, by the $t_z$-validity of $u_{z'}$ and V5, it would hold $\alpha \in u_{z'}$. Consequently, $F_{i'}^{u_{x'}}(x) \neq F_{i'}^{u_{x'}}(x)$. Hence, there exists a query $\beta$ that is asked by both $F_{i'}^{u_{x'}}(x)$ and $F_{i'}^{u_{x'}}(x)$ and that is in $u_{z} \cup u_{z'}$ (otherwise, both computations would output the same word). By definition of $Q(U_z)$ and $Q(U_{z'})$, it holds $\beta \in Q(U_z) \cap Q(U_{z'})$. Hence, $Q(U_z)$ and $Q(U_{z'})$ conflict in $\beta$ and $|\beta| \leq |x'| + i' < |c(i', x, y)| = |\alpha|$, in contradiction to the assumption that $\alpha$ is the least word which $Q(U_z)$ and $Q(U_{z'})$ conflict in.
In both cases we obtain a contradiction. Thus, the proof is complete. □

We want to show next that for all odd \( z \in \Sigma^n \) and all even \( z' \in \Sigma^n \) the sets \( Q(U_z) \) and \( Q(U_{z'}) \) indeed conflict. For the proof of this we need three more claims. We will make use of the next claim several times. In some cases a weaker version of this claim is sufficient. For better readability, we formulate this weaker statement in a separate claim (Claim 3.12).

**Claim 3.11** Let \( t = t_{s'} \) for some \( 0 \leq s' \leq s \) and \( z, z' \in \Sigma^n \) such that \( Q(U_z) \) and \( Q(U_{z'}) \) do not conflict. For each \( t \)-valid oracle \( v \supseteq u \) that is defined for exactly the words of length \( \leq n \) and that satisfies \( v(q) = u_z(q) \) for all \( |v| > q \in Q(U_z) \) and \( v(q) = u_{z'}(q) \) for all \( |v| > q \in Q(U_{z'}) \), there exists a \( t \)-valid oracle \( v' \supseteq v \) with \( |v'| = |u_z| \) and \( v'(q) = u_z(q) \) for all \( q \in Q(U_z) \), and \( v'(q) = u_{z'}(q) \) for all \( q \in Q(U_{z'}) \).

The following claim follows immediately from Claim 3.11 when we choose \( z = z' \) and \( s' = s \) (trivially, for no \( z \in \Sigma^n \) the set \( Q(U_z) \) conflicts with itself).

**Claim 3.12** Let \( z \in \Sigma^n \). For each \( t_s \)-valid oracle \( v \supseteq u \) that is defined for exactly the words of length \( \leq n \) and that satisfies \( v(q) = u_z(q) \) for all \( |v| > q \in Q(U_z) \), there exists a \( t_s \)-valid oracle \( v' \supseteq v \) with \( |v'| = |u_z| \) and \( v'(q) = u_z(q) \) for all \( q \in Q(U_z) \).

**Proof of Claim 3.11** Let \( w \supseteq v \) with \( |w| < |u_z| \), \( w(q) = u_z(q) \) for all \( |w| > q \in Q(U_z) \), and \( w(q) = u_{z'}(q) \) for all \( |w| > q \in Q(U_{z'}) \). Moreover, let \( \alpha = |w| \), i.e., \( \alpha \) is the least word that \( w \) is not defined for. It suffices to show the following:

- If \( \alpha = \emptyset' w^\kappa \) for some \( p' \in \mathbb{P}_3 \) with \( -p' \in \text{ran}(t) \) and \( \kappa > 0 \), then there exists a \( t \)-valid \( w' \supseteq w \) that is defined for the words of length \( p'^\kappa \), undefined for all words of greater length, and that satisfies \( w'(q) = u_z(q) \) for all \( |w'| > q \in Q(U_z) \) and \( w'(q) = u_{z'}(q) \) for all \( |w'| > q \in Q(U_{z'}) \).

  Note that in this case \( |w'| \leq |u_z| \) since \( u_z \) is defined for exactly the words of length \( \leq \gamma(n) \).

- If for all \( p' \in \mathbb{P}_3 \) with \( -p' \in \text{ran}(t) \) and all \( \kappa > 0 \) the word \( \alpha \) is not of length \( p'^\kappa \), then there exists \( b \in \{0, 1\} \) such that \( wb \) is \( t \)-valid, \( wb(q) = u_z(q) \) for all \( |wb| > q \in Q(U_z) \) and \( wb(q) = u_{z'}(q) \) for all \( |wb| > q \in Q(U_{z'}) \).

We study three cases.

1. Assume \( \alpha = \emptyset' w^\kappa \) for some \( p' \in \mathbb{P}_3 \) with \( -p' \in \text{ran}(t_s) \) and \( \kappa > 0 \). Then we let \( w' \supseteq w \) be the minimal oracle that is defined for all words of length \( p'^\kappa \) and contains \( 1'w^\kappa \), i.e., \( w' = w \cup \{1'w^\kappa\} \) when interpreting the oracles as sets. As \( u_z \cap \Sigma^p w^\kappa = u_{z'} \cap \Sigma^p w^\kappa = \{1'w^\kappa\} \) by Claim 3.8.2, we obtain \( w'(q) = u_z(q) \) for all \( |w'| > q \in Q(U_z) \) and \( w'(q) = u_{z'}(q) \) for all \( |w'| > q \in Q(U_{z'}) \). Moreover, if \( -p' \in \text{ran}(t) \), then \( w' \) is \( t \)-valid by Claim 3.7.2b and Claim 3.7.2c. If \( -p' \notin \text{ran}(t) \), then \( w' \) is \( t \)-valid by Claim 3.7.2a.

2. Now assume that \( \alpha = c(i', x, y) \) for \( i' \in \mathbb{N}^+ \) and \( x, y \in \mathbb{N} \) with \( 0 < t_s(i') \leq \alpha \). Let us first assume that \( \alpha \notin Q(U_z) \cup Q(U_{z'}) \). Then there exists \( b \in \{0, 1\} \) such that \( wb \) is \( t \)-valid (cf. Claim 3.7.2) and clearly \( wb(q) = u_z(q) \) for all \( |wb| > q \in Q(U_z) \) and \( wb(q) = u_{z'}(q) \) for all \( |wb| > q \in Q(U_{z'}) \).

  From now on we assume \( \alpha \in Q(U_z) \cup Q(U_{z'}) \). By symmetry, it suffices to consider the case \( \alpha \in Q(U_z) \). We study two cases.

  (a) If \( \alpha \in u_z \), then by \( V_1 \), \( F_p^{u_z}(x) = y \in K^{u_z} \). As all queries \( q \) of \( F_p^{u_z}(x) \) are in \( Q(U_z) \) and \( \alpha \notin Q(U_z) \cup Q(U_{z'}) \), it holds \( F_p^{u_z}(x) = F_p^{u_z}(x) = y \). Similarly,
we obtain \( y \in K^w \): If \( y = (0^{x''}, 0|z''|^{i''}, x') \) for \( i'' > 0 \) and \( x' \in \mathbb{N} \), then by \( y \in K^u \) the computation \( F_{v^s}^w(x') \) has an accepting path and all queries \( q \) of the least accepting path of this computation are in \( Q(U_z) \) and due to \( |q| \leq |y| < |\alpha| \) satisfy \( u_z(q) = w(q) \). Hence, \( F_{v^s}^w(x') \) accepts and \( y \in K^w \). Let us choose \( b = 1 \). Note that \( t(i') \) is not necessarily defined. If \( t(i') \) is defined, then \( t(i') = t_s(i') \) and we can apply Claim 3.7.2e and obtain that \( wb \) is \( t \)-valid. If \( t(i') \) is undefined, then we can apply Claim 3.7.2f and obtain that \( wb \) is \( t \)-valid. Clearly \( wb(q) = u_z(q) \) for all \( |wb| > q \in Q(U_z) \). In order to see that also \( wb(q) = u_z(q) \) for all \( |wb| > q \in Q(U_{z'}) \), it is sufficient to show that \( (\alpha \in Q(U_{z'}) \Rightarrow \alpha \in u_z) \). But this holds since otherwise, \( Q(U_z) \) and \( Q(U_{z'}) \) conflict.

(b) Assume \( \alpha \notin u_z \). Then by V5, \( F_{v^s}^u(x) \neq y \). As all queries \( q \) of \( F_{v^s}^u(x) \) are in \( Q(U_z) \) and due to \( |q| \leq |x| \), \( i < \alpha \) satisfy \( u_z(q) = w(q) \), it holds \( F_{v^s}^u(x) = F_{v^s}^u(x) \neq y \). Choose \( b = 0 \). Then by Claim 3.7.2g, \( wb \) is \( t \)-valid and clearly \( u_z(q) = wb(q) \) for all \( |wb| > q \in Q(U_z) \). In order to see \( u_z(q) = wb(q) \) for all \( |wb| > q \in Q(U_{z'}) \), it suffices to argue for \( \alpha \). If \( \alpha \in Q(U_{z'}) \), then \( \alpha \notin u_z \) as otherwise, \( Q(U_z) \) and \( Q(U_{z'}) \) would conflict.

3. We now consider the remaining cases, i.e., we may assume

- \( \alpha \) is not of length \( p''n \) for all \( p' \in \mathbb{P}_3 \) with \( -p' \in \text{ran}(t_s) \) and all \( n > 0 \) and
- \( \alpha \neq c(i', x, y) \) for all \( i' \in \mathbb{N}^+ \) and \( x, y \in \mathbb{N} \) with \( 0 < t_s(i') \leq \alpha \).

In this case, it holds \( \alpha \notin u_z \cup u_{z'} \) by Claim 3.8.1. We choose \( b = 0 \) and obtain that \( wb(q) = u_z(q) \) for all \( |wb| > q \in Q(U_z) \) and \( wb(q) = u_z(q) \) for all \( |wb| > q \in Q(U_{z'}) \). Moreover, by Claim 3.7.2, \( wb \) is \( t \)-valid.

This finishes the proof of Claim 3.11.

\[ \square \]

Claim 3.13 For all \( z \in \Sigma^n \) it holds \( z \in Q(U_z) \).

Proof For a contradiction, assume \( z \notin Q(U_z) \) for some \( z \in \Sigma^n \). We study the cases \( i = j \) and \( i \neq j \) separately.

First assume \( i = j \). In this case \( p \in \mathbb{P}_1 \). Let \( u' \) be the oracle that is defined for exactly the words of length \( \leq n \) and satisfies \( u' = u \) when the oracles are considered as sets. Then \( u' \) is \( t_s \)-valid by Claim 3.7.2d and \( u' \) and \( u_z \) agree on all words in \( \Sigma^n \cap Q(U_z) \) as \( u_z \cap \Sigma^n = \{ z \} \) and \( z \notin Q(U_z) \). Thus, we can apply Claim 3.12 to the oracle \( u' \). Hence, there exists a \( t_s \)-valid oracle \( v \) satisfying \( |v| = |u_z|, v \cap \Sigma^n = \emptyset \), and \( v(q) = u_z(q) \) for all \( q \in Q(U_z) \). By the latter property and the fact that \( U_z \subseteq Q(U_z) \) contains all queries asked by the least accepting path of \( M_v^\\alpha(F_v^u(0^n)) \), this path is also an accepting path of the computation \( M_v^\\alpha(F_v^u(0^n)) \). As \( v \) is defined for all words of length \( \leq \gamma(n) \), the computation \( M_v^\\alpha(F_v^u(0^n)) \) is defined. Thus, \( 0^n \notin C_v \) for all \( v' \supseteq v \) and \( M_v^\\alpha(F_v^u(0^n)) \) definitely accepts, in contradiction to the assumption that step \( s \) of the construction fails.

Now let us consider the case \( i \neq j \). Here \( p \in \mathbb{P}_3 \). By symmetry, it suffices to consider the case that \( z \) is odd. Let \( z' \) be the minimal even element of \( \Sigma^n \) that is not in \( Q(U_z) \). Such \( z' \) exists as it holds \( 2^{n-1} > 4\gamma(n) > 2\gamma(n) \) by (5), \( \ell(Q(U_z)) \leq 2\gamma(n) \) by Claim 3.9, and hence, \( \ell(Q(U_z)) \leq 2\gamma(n) < 2^{n-1} = \{ z'' \in \Sigma^n \mid z'' \text{ even} \} \). Now choose \( u' \) to be the oracle that is defined for exactly the words of length \( \leq n \) and that satisfies \( u' = u \cup \{ z' \} \) when the oracles are considered as sets. Then \( u' \) is \( t_s \)-valid by Claim 3.7.2b and Claim 3.7.2g. Moreover, as \( z', z' \notin Q(U_z) \), the oracles \( u' \) and \( u_z \) agree on all words in \( \Sigma^n \cap Q(U_z) \). Thus, we can apply Claim 3.12 to the oracle \( u' \) for the parameter \( z \) and obtain a \( t_s \)-valid oracle \( v \) that is defined for all words of length \( \leq \gamma(n) \) and satisfies both \( v \cap \Sigma^n = \{ z' \} \) and \( v(q) = u_z(q) \) for all \( q \in Q(U_z) \). The latter property and the fact that \( U_z \subseteq Q(U_z) \) contains all queries asked by the least accepting path of \( M_v^\\alpha(F_v^u(0^n)) \)
yield that this path is also an accepting path of the computation \( M^u_i(F^u_r(0^n)) \). As \( v \) is defined for all words of length \( \leq \gamma(n) \), the computation \( M^u_i(F^u_r(0^n)) \) definitely accepts. Let us study two cases depending on whether \( M^u_i(F^u_r(0^n)) \) definitely accepts or definitely rejects (note that this computation is defined as \( v \) is defined for all words of length \( \leq \gamma(n) \));

- Assume that \( M^u_j(F^u_r(0^n)) \) definitely accepts. Let \( s' \) be the step that treats the task \((i, j)\). Hence, \( s' < s \) since \( t_s(i, j) \) is defined. By Claim 3.5, the oracle \( v \) is \( t'_s - 1 \)-valid. Now, as both \( M^u_i(F^u_r(0^n)) \) and \( M^u_j(F^u_r(0^n)) \) definitely accept, \( v \) is even \( t'' \)-valid for \( t'' = t'_s - 1 \cup \{(i, j) \mapsto 0\} \). But then the construction would have chosen \( t_s = t'' \), in contradiction to \( t_s(i, j) \neq 0 \).

- Assume that \( M^u_j(F^u_r(0^n)) \) definitely rejects. As \( v \cap \Sigma^n = \{z'\} \), it holds \( 0^n \in B^u_p \) for all \( v' \supseteq v \). This is a contradiction to the assumption that step \( s \) of the construction fails.

As in both cases we obtain a contradiction, the proof of Claim 3.13 is complete. \( \square \)

**Claim 3.14** For all odd \( z \in \Sigma^n \) and all even \( z' \in \Sigma^n \), \( Q(U_z) \) and \( Q(U_{z'}) \) conflict.

**Proof** Assume there are \( z \) odd and \( z' \) even such that \( Q(U_z) \) and \( Q(U_{z'}) \) do not conflict. Then let \( u' \supseteq u \) be the minimal oracle that is defined for all words of length \( \leq n \) and contains \( z \) and \( z' \), i.e., interpreting oracles as sets it holds \( u' = u \cup \{z, z'\} \). Let \( s' \) be the step that treats the task \((i, j)\). Then \( s' < s \) as \( t_s(i, j) \) is defined. As \( t_s \in T \) is injective on its support and \( t_s(i, j) = -p \), it holds \( -p \notin \text{ran}(t_{s'-1}) \). Therefore, the oracle \( u' \) is \( t_{s'-1} \)-valid by Claim 3.7.a. If Claim 3.11 cannot be applied to the oracle \( u' \) for the parameters \( z, z' \), and \( s' - 1 \), then \( z \in Q(U_z) \) or \( z' \in Q(U_{z'}) \). As by Claim 3.13, \( z \in Q(U_z) \) and \( z' \in Q(U_{z'}) \) and moreover, \( u_z \cap \Sigma^n = \{z\} \) and \( u_{z'} \cap \Sigma^n = \{z'\} \), in this case \( Q(U_z) \) and \( Q(U_{z'}) \) conflict, a contradiction. Hence, it remains to consider the case that Claim 3.11 can be applied to the oracle \( u' \) for the parameters \( z, z' \), and \( s' - 1 \).

Applying Claim 3.11, we obtain a \( t_{s'-1} \)-valid \( v \supseteq u' \) that is defined for all words of length \( \leq \gamma(n) \) and that satisfies \( v(q) = u_z(q) \) for all \( q \in Q(U_z) \) and \( v(q) = u_{z'}(q) \) for all \( q \in Q(U_{z'}) \).

We claim

\[
v \text{ is } t''\text{-valid for } t'' = t_{s'-1} \cup \{(i, j) \mapsto 0\}. \tag{6}
\]

Once (6) is proven, we obtain a contradiction as then the construction would have chosen \( t_s = t'' \), in contradiction to \( t_s(i, j) \neq 0 \). Hence, then our assumption is wrong and for all odd \( z \in \Sigma^n \) and all even \( z' \in \Sigma^n \), \( Q(U_z) \) and \( Q(U_{z'}) \) conflict.

It remains to prove (6). We study two cases.

**Case 1:** first we assume that \( i \neq j \), i.e., it suffices to prove that \( M^u_i(F^u_r(0^n)) \) and \( M^u_j(F^u_r(0^n)) \) definitely accept. Recall that \( M^u_i(F^u_r(0^n)) \) and \( M^u_j(F^u_r(0^n)) \) definitely accept. Moreover, \( v(q) = u_z(q) \) for all \( q \in Q(U_z) \) and \( v(q) = u_{z'}(q) \) for all \( q \in Q(U_{z'}) \) and in particular, \( v \) is defined for all words in \( Q(U_z) \cup Q(U_{z'}) \). This implies that the least accepting paths of \( M^u_i(F^u_r(0^n)) \) and \( M^u_j(F^u_r(0^n)) \) are also accepting paths of the computations \( M^u_i(F^u_r(0^n)) \) and \( M^u_j(F^u_r(0^n)) \). Thus, \( v \) is \( t'' \)-valid.

**Case 2:** assume that \( i = j \), i.e., we have to prove that on some input \( x \) the computation \( M^u_i(x) \) has two accepting paths. By Claim 3.13, \( z \in Q(U_z) \) and \( z' \in Q(U_{z'}) \). As \( Q(U_z) \) and \( Q(U_{z'}) \) do not conflict, it holds \( z \notin \text{ran}(t_{s'}) \). This implies \( Q(U_z) \neq Q(U_{z'}) \). Let \( \kappa \in \mathbb{N} \) be minimal such that \( Q_\kappa(U_z) \neq Q_\kappa(U_{z'}) \) and for a contradiction, assume \( \kappa > 0 \).

Let \( \alpha \in Q_\kappa(U_z) \Delta Q_\kappa(U_{z'}) \). Without loss of generality, we assume \( \alpha \in Q_\kappa(U_z) - Q_\kappa(U_{z'}) \). Then there exist \( i', x, y \in \mathbb{N} \) with \( i' > 0 \) such that \( c(i', x, y) \in Q_{\kappa-1}(U_z) \) and \( F^u_{\alpha}(x) \) asks the query \( \alpha \). By the choice of \( \kappa \), it holds \( Q_{\kappa-1}(U_{z'}) = Q_{\kappa-1}(U_z) \) and thus, \( c(i', x, y) \in Q_{\kappa-1}(U_{z'}) \).

17
Consequently, all queries of \( F_{\beta}^u(x) \) are in \( Q_\kappa(U_{\beta'}) \). However, \( \alpha \notin Q_\kappa(U_{\beta'}) \) and therefore, \( \alpha \) cannot be asked by \( F_{\beta}^u(x) \). This shows that there is a word \( \beta \in u_\triangle U_{\beta'} \) asked by both \( F_{\beta}^u(x) \) and \( F_{\beta'}^u(x) \) (otherwise, the two computations would ask the same queries). But then \( \beta \in Q_\kappa(U) \cap Q_\kappa(U_{\beta'}) \), which implies that \( Q(U) \) and \( Q(U_{\beta'}) \) conflict, a contradiction. Hence, we obtain \( \kappa = 0 \) and \( U_0 = Q_0(U) = U_{\beta'} \).

Recall that \( U_\beta \) (resp., \( U_{\beta'} \)) is the set consisting of all oracle queries of the least accepting path \( P \) (resp., \( P' \)) of the construction \( M_\beta^u(F_\beta^u(0^n)) \) (resp., \( M_{\beta'}^u(F_{\beta'}^u(0^n)) \)). As \( u_\beta(q) = v(q) \) for all \( q \in Q(U_\beta) \supseteq U_\beta \) and \( u_{\beta'}(q) = v(q) \) for all \( q \in Q(U_{\beta'}) \supseteq U_{\beta'} \), the paths \( P \) and \( P' \) are accepting paths of the computation \( M_\beta^u(F_\beta^u(0^n)) \). Finally, \( P \) and \( P' \) are distinct paths since \( U_\beta \) and \( U_{\beta'} \) are distinct sets. This finishes the proof of (6). Hence, the proof of Claim 3.14 is complete.

The remainder of the proof that the construction is possible is based on an idea by Hartmanis and Hemachandra [HH88]. Consider the set

\[
E = \{ \{z, z'\} \mid z, z' \in \Sigma^n, z \text{ odd } \iff z' \text{ even}, (z \in Q(U_{\beta'}) \lor z' \in Q(U_{\beta})) \}
\]

\[
= \bigcup_{z \in \Sigma^n} \{ \{z, z'\} \mid z, z' \in \Sigma^n, z \text{ odd } \iff z' \text{ even}, z' \in Q(U_{\beta}) \}. \tag{7}
\]

Let \( z, z' \in \Sigma^n \) such that \( (z \text{ odd } \iff z' \text{ even}) \). Then by Claim 3.14 and Claim 3.10, \( Q(U_{\beta'}) \) and \( Q(U_{\beta'}) \) conflict in a word of length \( n \). As observed above, this means that they conflict in \( z \) or \( z' \). Hence, \( z \in Q(U_{\beta'}) \) or \( z' \in Q(U_{\beta'}) \). This shows \( E = \{ \{z, z'\} \mid z, z' \in \Sigma^n, z \text{ odd } \iff z' \text{ even} \} \) and thus, \(|E| = 2^{2n-2} \). By Claim 3.9, for each \( z \in \Sigma^n \) it holds \(|Q(U_{\beta'})| \leq (\ell(Q(U_{\beta'})) \leq 2\gamma(n)\). Consequently,

\[
|E| \leq \sum_{z \in \Sigma^n} |Q(U_{\beta'})| |\leq 2^n \cdot 2\gamma(n) = 2^{n+1} \cdot \gamma(n) < 2^{2^{n-2}} = |E|,
\]

a contradiction. Hence, the assumption that the construction fails in step \( s \) treating the task \((i, j, r)\) is wrong. This shows that the construction described above is possible and \( O \) is well-defined. In order to finish the proof of the Theorem 3.2, it remains to show that

- \( \text{DisjNP}^O \) does not contain a pair \( \leq_m \text{NP}^O \)-hard for \( \text{NP}^O \cap \text{coNP}^O \),
- each problem in \( \text{NP}^O \) has a \( \text{PO} \)-optimal proof system, and
- \( \text{UP} \) does not contain a \( \leq_m \text{NP} \)-complete problem.

**Claim 3.15** \( \text{DisjNP}^O \) does not contain a pair that is \( \leq_m \text{NP} \)-hard for \( \text{NP}^O \cap \text{coNP}^O \).

**Proof** Assume the assertion is wrong, i.e., there exist distinct \( i, j \in \mathbb{N}^+ \) such that \((L(M_i^O), L(M_j^O)) \in \text{DisjNP}^O \) and for every \( A \in \text{NP}^O \cap \text{coNP}^O \) it holds \( A \leq_m \text{NP}^O (L(M_i^O), L(M_j^O)) \). From \( L(M_i^O) \cap L(M_j^O) = \emptyset \) it follows that for all \( s \) there does not exist \( z \) such that both \( M_i^u(z) \) and \( M_j^u(z) \) definitely accept. Hence, for no \( s \) it holds \( t_s(i, j) = 0 \) and thus, by construction \( t_s(i, j) = -p \) for some \( p \in \mathbb{P}_3 \) and all sufficiently large \( s \). The latter implies \(|O \cap \Sigma^p \{\} = 1 \) for all \( k > 0 \) (cf. V3), which yields \( A_p = \overline{B_p} \), i.e., \( A_p \in \text{NP}^O \cap \text{coNP}^O \). Thus, there exists \( r \) such that \( A_p \leq_m \text{NP}^O (L(M_i^O), L(M_j^O)) \) via \( F_r^O \). Let \( s \) be the step that treats task \((i, j, r)\). This step makes sure that there exists \( n \in \mathbb{N}^+ \) such that at least one of the following properties holds:

- \( 0^n \in A_p \) for all \( v \sqcup w_s \) and \( M_i^u(F_{i}^u(0^n)) \) definitely rejects,
- \( 0^n \in B_p \) for all \( v \sqcup w_s \) and \( M_j^u(F_{j}^u(0^n)) \) definitely rejects.

18
As $O(q) = w_s(q)$ for all $q$ that $w_s$ is defined for, one of the following two statements holds.

- $0^n \in A_p^O$ and $F^O(0^n)$ is rejected by $M_i^O$.
- $0^n \in B_p^O = \overline{A_p^O}$ and $F^O_r(0^n)$ is rejected by $M_j^O$.

This is a contradiction to $A_p^O \leq_{m}^O (L(M_i^O), L(M_j^O))$ via $F^O_r$, which completes the proof of Claim 3.15.

Claim 3.16 Each problem in NP$^O$ has a PO-optimal proof system.

Proof By Corollary 2.4, it suffices to prove that K$^O$ has a PO-optimal proof system.

Let $g \in FP^O$ be an arbitrary proof system for K$^O$ and $a$ be an arbitrary element of K$^O$. Define $f$ to be the following function $\Sigma^* \rightarrow \Sigma^*$:

$$f(z) = \begin{cases} g(z') & \text{if } z = 1z' \\ y & \text{if } z = 0c(i, x, y) \text{ for } i \in \mathbb{N}^+, x, y \in \mathbb{N}, \text{ and } c(i, x, y) \in O \\ a & \text{otherwise} \end{cases}$$

By definition, $f \in FP^O$ and as $g$ is a proof system for K$^O$ it holds $f(\Sigma^*) \supseteq K^O$. We show $f(\Sigma^*) \subseteq K^O$. Let $z \in \Sigma^*$. Assume $z = 0c(i, x, y)$ for $i \in \mathbb{N}^+, x, y \in \mathbb{N}$, and $c(i, x, y) \in O$ (otherwise, clearly $f(z) \notin K^O$). Let $j > 0$ such that $F_j^O$ computes $f$. Let $s$ be large enough such that $w_s$ is defined for $c(i, x, y)$, i.e., $w_s(c(i, x, y)) = 1$. As $w_s$ is $t_s$-valid, we obtain by V1 that $F_{w_s}(x) = \pi(x) \in K^{w_s}$ and by Claim 3.4 that $F_{w_s'}(x)$ is defined and $z \in K^O$ for all $v \supseteq w_s$. Then $F_j^O(x) \notin K^O$. This shows that $f$ is a proof system for K$^O$.

It remains to show that each proof system for K$^O$ is PO-simulated by $f$. Let $h$ be an arbitrary proof system for K$^O$. Then there exists $i > 0$ such that $F_i^O$ computes $h$. By construction, $t_s(i) = 0$, where $s$ is the number of the step that treats the task $i$. Consider the following function $\pi : \Sigma^* \rightarrow \Sigma^*$:

$$\pi(x) = \begin{cases} 0c(i, x, F_i^O(x)) & \text{if } c(i, x, F_i^O(x)) \geq t_s(i) \\ z & \text{if } c(i, x, F_i^O(x)) < t_s(i) \text{ and } z \text{ is minimal with } f(z) = F_i^O(x) \end{cases}$$

As $f$ and $F_i^O$ are proof systems for K$^O$, for every $x$ there exists $z$ with $f(z) = F_i^O(x)$. Hence, $\pi$ is total. Since $t_s(i)$ is a constant, $\pi \in FP \subseteq FP^O$. It remains to show that $f(\pi(x)) = F_i^O(x)$ for all $x \in \Sigma^*$. If $|x| < m$, it holds $f(\pi(x)) = F_i^O(x)$. Otherwise, choose $s'$ large enough such that (i) $t_{s'}(i)$ is defined (i.e., $t_{s'}(i) = t_s(i)$) and (ii) $w_{s'}$ is defined for $c(i, x, F_{w_{s'}}^O(x))$. Then, as $w_{s'}$ is $t_{s'}$-valid, V5 yields that $c(i, x, F_{w_{s'}}^O(x)) \in w_{s'}$. By Claim 3.4, $F^O_{w_{s'}}(x)$ is defined and hence, $F_i^O(x) = F^O_{w_{s'}}(x)$ as well as $c(i, x, F_i^O(x)) \in w_{s'} \subseteq O$. Hence, $f(\pi(x)) = F_i^O(x)$, which shows $h = F_i^O \leq_{m}^{PO} f$. This completes the proof of Claim 3.16.

Claim 3.17 UP$^O$ does not contain a $\leq_{m}^{PO}$-complete problem.

Proof Assume there exists an UP$^O$-complete problem. Then there exists $i > 0$ such that $L(M_i^O)$ is $\leq_{m}^{PO}$-complete for UP$^O$. As on every input, $M_i^O$ has at most one accepting path, there exists no $s > 0$ with $t_s(i, i) = 0$. Hence, by construction $t_s(i, i) = -q$ for some $q \in \mathbb{F}_1$ and all sufficiently large $s$. Then $|O \cap \Sigma^+| \leq 1$ for all $k > 0$ (cf. V7) and consequently, $C_q^O \in UP^O$. As $L(M_i^O)$ is complete for UP$^O$, there exists $r > 0$ such that $C_q^O \leq_{m}^{PO} L(M_i^O)$ via $F_r^O$. Let $s > 0$ be the step that treats the task $(i, i, r)$. By construction, there exists $n \in \mathbb{N}^+$ such that one of the following two statements holds:
\[0^n \in C^v_q \text{ for all } v \sqsupseteq w_s \text{ and } M_i^{w_s}(F_r^{w_s}(0^n)) \text{ definitely rejects.}\]

\[0^n \notin C^v_q \text{ for all } v \sqsupseteq w_s \text{ and } M_i^{w_s}(F_r^{w_s}(0^n)) \text{ definitely accepts.}\]

As \(O\) and \(w_s\) agree on all words that \(w_s\) is defined for, one of the following two conditions holds:

\[0^n \in C^O_q \text{ and } M_i^O(F_r^O(0^n)) \text{ rejects.}\]

\[0^n \notin C^O_q \text{ and } M_i^O(F_r^O(0^n)) \text{ accepts.}\]

This is a contradiction to \(C^O_q \leq_{\text{m}}^O L(M_i^O)\) via \(F_r^O\), which shows that \(\text{UP}^O\) does not have \(\leq_{\text{m}}^O\)-complete problems. This completes the proof of Claim 3.17.

Now the proof of Theorem 3.2 is complete.

\[\square\]

References

[BKM09] O. Beyersdorff, J. Köbler, and J. Messner. Nondeterministic functions and the existence of optimal proof systems. \textit{Theor. Comput. Sci.}, 410(38-40):3839–3855, 2009.

[CR79] S. Cook and R. Reckhow. The relative efficiency of propositional proof systems. \textit{Journal of Symbolic Logic}, 44:36–50, 1979.

[DG19] T. Dose and C. Glaßer. \(\text{NP}\)-completeness, proof systems, and disjoint \(\text{NP}\)-pairs. Technical Report 19-050, Electronic Colloquium on Computational Complexity (ECCC), 2019.

[Dos19] T. Dose. \(P \neq \text{NP}\) and all sets in \(\text{NP} \cup \text{coNP}\) have \(P\)-optimal proof systems relative to an oracle. \textit{arXiv e-prints}, page arXiv:1909.02839, Sep 2019.

[ESY84] S. Even, A. L. Selman, and J. Yacobi. The complexity of promise problems with applications to public-key cryptography. \textit{Information and Control}, 61:159–173, 1984.

[EY80] S. Even and Y. Yacobi. Cryptocomplexity and \(\text{NP}\)-completeness. In \textit{Proceedings 7th International Colloquium on Automata, Languages and Programming}, volume 85 of \textit{Lecture Notes in Computer Science}, pages 195–207. Springer, 1980.

[GS88] J. Grollmann and A. L. Selman. Complexity measures for public-key cryptosystems. \textit{SIAM Journal on Computing}, 17(2):309–335, 1988.

[GSSZ04] C. Glaßer, A. L. Selman, S. Sengupta, and L. Zhang. Disjoint \(\text{NP}\)-pairs. \textit{SIAM Journal on Computing}, 33(6):1369–1416, 2004.

[HH88] J. Hartmanis and L. A. Hemachandra. Complexity classes without machines: On complete languages for \(\text{UP}\). \textit{Theor. Comput. Sci.}, 58:129–142, 1988.

[Kha19] E. Khaniki. New relations and separations of conjectures about incompleteness in the finite domain. \textit{arXiv e-prints}, pages 1–25, Apr 2019.

[KM00] J. Köbler and J. Messner. Is the standard proof system for sat \(p\)-optimal? In S. Kapoor and S. Prasad, editors, \textit{FSTTCS 2000: Foundations of Software Technology and Theoretical Computer Science}, pages 361–372, Berlin, Heidelberg, 2000. Springer Berlin Heidelberg.
[KMT03] J. Köbler, J. Messner, and J. Torán. Optimal proof systems imply complete sets for promise classes. *Information and Computation*, 184(1):71–92, 2003.

[KP89] J. Krajíček and P. Pudlák. Propositional proof systems, the consistency of first order theories and the complexity of computations. *Journal of Symbolic Logic*, 54:1063–1079, 1989.

[MP91] N. Megiddo and C. H. Papadimitriou. On total functions, existence theorems and computational complexity. *Theor. Comput. Sci.*, 81(2):317–324, 1991.

[Pap94] C. M. Papadimitriou. *Computational complexity*. Addison-Wesley, Reading, Massachusetts, 1994.

[Pud96] P. Pudlák. On the lengths of proofs of consistency. In *Collegium Logicum*, pages 65–86. Springer Vienna, 1996.

[Pud13] P. Pudlák. *Logical Foundations of Mathematics and Computational Complexity - A Gentle Introduction*. Springer monographs in mathematics. Springer, 2013.

[Pud17] P. Pudlák. Incompleteness in the finite domain. *The Bulletin of Symbolic Logic*, 23(4):405–441, 2017.

[Raz94] A. A. Razborov. On provably disjoint np-pairs. *Electronic Colloquium on Computational Complexity (ECCC)*, 1(6), 1994.