Decision Problems For Turing Machines
Olivier Finkel, Dominique Lecomte

To cite this version:
Olivier Finkel, Dominique Lecomte. Decision Problems For Turing Machines. Information Processing Letters, 2009, 109 (23-24), pp.1223-1226. hal-00413331

HAL Id: hal-00413331
https://hal.science/hal-00413331
Submitted on 3 Sep 2009

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L’archive ouverte pluridisciplinaire HAL, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d’enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.
Abstract

We answer two questions posed by Castro and Cucker in [CC89], giving the exact complexities of two decision problems about cardinalities of \( \omega \)-languages of Turing machines. Firstly, it is \( D_2(\Sigma^1_1) \)-complete to determine whether the \( \omega \)-language of a given Turing machine is countably infinite, where \( D_2(\Sigma^1_1) \) is the class of 2-differences of \( \Sigma^1_1 \)-sets. Secondly, it is \( \Sigma^1_1 \)-complete to determine whether the \( \omega \)-language of a given Turing machine is uncountable.

Keywords. Theory of computation; computational complexity; formal languages; \( \omega \)-languages; Turing machines; decision problems; analytical hierarchy.

1 Introduction

Many classical decision problems arise naturally in the fields of Formal Language Theory and of Automata Theory. Castro and Cucker studied decision problems for \( \omega \)-languages of Turing machines in [CC89]. Their motivation was, on the one hand, to classify problems about Turing machines and, on the other hand, to “give natural complete problems for the lowest levels of the analytical hierarchy which
constitute an analog of the classical complete problems given in recursion theory for the arithmetical hierarchy.” They studied the degrees of many classical decision problems like: “Is the ω-language recognized by a given machine non empty?”, “Is it finite?” “Do two given machines recognize the same ω-language?” In particular, they proved that the non-emptiness and the infiniteness problems for ω-languages of Turing machines are Σ₁¹-complete, and that the universality problem, the inclusion problem, and the equivalence problem are Π₁²-complete. Thus these problems are located at the first or the second level of the analytical hierarchy and are “highly undecidable.”

Notice that Staiger studied also in [Sta93] the verification property, which is in fact the inclusion problem, for many classes of ω-languages located at one of the first three levels of the arithmetical hierarchy. Cenzer and Remmel studied in [CR03] some decision problems for classes of ω-languages accepted by some computable deterministic automata. These classes of ω-languages are located at one of the first three levels of the arithmetical hierarchy, while the class of ω-languages of Turing machines considered by Castro and Cucker and in this paper is actually the class of effective analytic sets. Thus the class we consider in this paper is much larger than the classes studied by Cenzer and Remmel in [CR03]. Cenzer and Remmel studied also various approximate verification properties, determining the index sets for pairs of languages (V,W) such that W − V is finite, is a set of measure zero or contains only finitely many computable sequences. The verification property is also studied by Klarlund in [Kla94].

The following questions were left open by Castro and Cucker in [CC89]. What is the complexity of the following decision problems: “Is the ω-language recognized by a given Turing machine countably infinite?”, “Is the ω-language recognized by a given Turing machine uncountable?” We answer here these questions, giving the exact complexities of these two decision problems about cardinalities of ω-languages of Turing machines. Firstly, it is $D_2(\Sigma_1^1)$-complete to determine whether the ω-language of a given Turing machine is countably infinite, where $D_2(\Sigma_1^1)$ is the class of 2-differences of $\Sigma_1^1$-sets. Secondly, it is $\Sigma_1^1$-complete to determine whether the ω-language of a given Turing machine is uncountable. This can be compared with this corresponding result of [CR03, CR99]. It is $\Pi_3^0$-complete to determine whether a given $\Pi_1^0$ ω-language is infinite. It is $\Sigma_1^1$-complete to determine whether a given $\Pi_1^0$ ω-language is uncountable, and it is $\Pi_1^1$-complete to determine whether a given $\Pi_1^0$ ω-language is countably infinite, see [CR99, Theorem 4.5]. We refer the reader to [CR03] for results about other classes of ω-languages, like the class of $\Sigma_1^0$ ω-languages, or the class of $\Pi_2^0$ ω-languages.
2 Recall of basic notions

The set of natural numbers is denoted by \( \mathbb{N} \). We assume the reader to be familiar with the arithmetical and analytical hierarchies on subsets of \( \mathbb{N} \), these notions may be found in the textbooks on computability theory \([\text{Rog67}]\) \([\text{Odi89}, \text{Odi99}]\).

We now recall the notions of 1-reduction and of \( \Sigma^1_1 \)-completeness (respectively, \( \Pi^1_1 \)-completeness). Given two sets \( A, B \subseteq \mathbb{N} \) we say \( A \) is 1-reducible to \( B \) and write \( A \leq_1 B \) if there exists a total computable injective function \( f \) from \( \mathbb{N} \) to \( \mathbb{N} \) with \( A = f^{-1}[B] \). A set \( A \subseteq \mathbb{N} \) is said to be \( \Sigma^1_n \)-complete (respectively, \( \Pi^1_n \)-complete) iff \( A \) is a \( \Sigma^1_n \)-set (respectively, \( \Pi^1_n \)-set) and for each \( \Sigma^1_n \)-set (respectively, \( \Pi^1_n \)-set) \( B \subseteq \mathbb{N} \) it holds that \( B \leq_1 A \). An important fact is that, for each integer \( n \geq 1 \), there exist some \( \Sigma^1_n \)-complete subset of \( \mathbb{N} \). Examples of such sets are precisely described in \([\text{Rog67}]\) or \([\text{CC89}]\). In the sequel \( E_1 \) denotes a \( \Sigma^1_1 \)-complete subset of \( \mathbb{N} \). The set \( E_1 = \mathbb{N} - E_1 \subseteq \mathbb{N} \) is a \( \Pi^1_1 \)-complete set.

We assume now the reader to be familiar with the theory of formal (\( \omega \))-languages \([\text{Tho90}, \text{Sta97}]\). We recall some usual notations of formal language theory.

When \( \Sigma \) is a finite alphabet, a non-empty finite word over \( \Sigma \) is any sequence \( x = a_1 \ldots a_k \), where \( a_i \in \Sigma \) for \( i = 1, \ldots, k \), and \( k \) is an integer \( \geq 1 \). \( \Sigma^* \) is the set of finite words (including the empty word) over \( \Sigma \).

The first infinite ordinal is \( \omega \). An \( \omega \)-word over \( \Sigma \) is an \( \omega \)-sequence \( a_1 \ldots a_n \ldots \), where for all integers \( i \geq 1 \), \( a_i \in \Sigma \). When \( \sigma \) is an \( \omega \)-word over \( \Sigma \), we write \( \sigma = \sigma(1)\sigma(2)\ldots \sigma(n)\ldots \), where for all \( i \), \( \sigma(i) \in \Sigma \).

The usual concatenation product of two finite words \( u \) and \( v \) is denoted \( u \cdot v \) and sometimes just \( uv \). This product is extended to the product of a finite word \( u \) and an \( \omega \)-word \( v \): the infinite word \( u \cdot v \) is then the \( \omega \)-word such that:

\[
(u \cdot v)(k) = u(k) \text{ if } k \leq |u|, \text{ and } (u \cdot v)(k) = v(k - |u|) \text{ if } k > |u|.
\]

The set of \( \omega \)-words over the alphabet \( \Sigma \) is denoted by \( \Sigma^\omega \). An \( \omega \)-language over an alphabet \( \Sigma \) is a subset of \( \Sigma^\omega \).

Recall now the notion of acceptance of infinite words by Turing machines considered by Castro and Cucker in \([\text{CC89}]\).

**Definition 2.1** A non deterministic Turing machine \( \mathcal{M} \) is a 5-tuple \( \mathcal{M} = (Q, \Sigma, \Gamma, \delta, q_0) \), where \( Q \) is a finite set of states, \( \Sigma \) is a finite input alphabet, \( \Gamma \) is a finite tape alphabet satisfying \( \Sigma \subseteq \Gamma \), \( q_0 \) is the initial state, and \( \delta \) is a mapping from \( Q \times \Gamma \) to subsets of \( Q \times \Gamma \times \{L, R, S\} \). A configuration of \( \mathcal{M} \) is a triple \( (q, \sigma, i) \), where \( q \in Q \), \( \sigma \in \Gamma^\omega \) and \( i \in \mathbb{N} \). An infinite sequence of configurations \( r = (q_i, \alpha_i, j_i)_{i \geq 1} \) is called a run of \( \mathcal{M} \) on \( w \in \Sigma^\omega \) iff:

\[
(a) \ (q_1, \alpha_1, j_1) = (q_0, w, 1), \quad \text{and}
\]
(b) for each $i \geq 1$, $(q_i, \alpha_i, j_i) \vdash (q_{i+1}, \alpha_{i+1}, j_{i+1})$,

where $\vdash$ is the transition relation of $M$ defined as usual. The run $r$ is said to be complete if $(\forall n \geq 1)(\exists k \geq 1)(j_k \geq n)$. The run $r$ is said to be oscillating if $(\exists k \geq 1)(\forall n \geq 1)(\exists m \geq n)(j_m = k)$.

**Definition 2.2** Let $M = (Q, \Sigma, \Gamma, \delta, q_0)$ be a non deterministic Turing machine and $F \subseteq Q$. The $\omega$-language accepted by $(M, F)$ is the set of $\omega$-words $\sigma \in \Sigma^\omega$ such that there exists a complete non oscillating run $r = (q_i, \alpha_i, j_i)_{i \geq 1}$ of $M$ on $\sigma$ such that, for all $i, q_i \in F$.

The above acceptance condition is denoted 1'-acceptance in [CG78]. Other usual acceptance conditions are the now called B"uchi or Muller acceptance conditions, respectively denoted 2-acceptance and 3-acceptance in [CG78]. Cohen and Gold proved the following result in [CG78, Theorem 8.2].

**Theorem 2.3 (Cohen and Gold [CG78])** An $\omega$-language is accepted by a non deterministic Turing machine with 1'-acceptance condition iff it is accepted by a non deterministic Turing machine with B"uchi (respectively, Muller) acceptance condition.

Notice that this result holds because Cohen’s and Gold’s Turing machines accept infinite words via complete non oscillating runs, while 1', B"uchi or Muller acceptance conditions refer to the sequence of states entered during an infinite run.

For other approaches, acceptance is based only on the sequence of states entered by the machine during an infinite computation [Sta97], or one requires also that the machine reads the whole infinite tape [EH93]. We refer the reader to [SW78, Sta99, FS00, Sta00] for a study of these different approaches.

We recall the existence of the arithmetical and analytical hierarchies of $\omega$-languages, see [SW78, Sta97]; see also [LT93] about logical specifications for infinite computations. The first class of the analytical hierarchy is the class $\Sigma_1^1$ of effective analytic sets which are obtained by projection of arithmetical sets. By [Sta99, Theorem 16] (see also [Sta00, Theorem 5.2]) we have the following characterization of the class of $\omega$-languages accepted by non deterministic Turing machines via acceptance by complete runs (i.e., not necessarily non oscillating).

**Theorem 2.4 ([Sta99])** The class of $\omega$-languages accepted by non deterministic Turing machines with 1' (respectively, B"uchi, Muller) acceptance condition is the class $\Sigma_1^1$ of effective analytic sets.

We return now to Cohen’s and Gold’s non deterministic Turing machines accepting via complete non oscillating runs. The following result follows from [CG78, Note 2 page 12] and from Theorem 2.4.
Theorem 2.5 The class of \(\omega\)-languages accepted by Cohen’s and Gold’s non deterministic Turing machines with \(1'\) (respectively, Büchi, Muller) acceptance condition is the class \(\Sigma^1_1\) of effective analytic sets.

3 Decision problems about Turing machines

In the sequel we consider, as in [CC89], that the alphabet \(\Sigma\) contains only two letters \(a\) and \(b\), and we shall denote \(M_z\) the non deterministic Turing machine of index \(z\), reading words over \(\Sigma\), equipped with a \(1'\)-acceptance condition. We now recall the results of Castro and Cucker giving the exact complexity of the non-emptiness problem and of the infiniteness problem for \(\omega\)-languages of Turing machines.

Theorem 3.1

1. \(\{z \in \mathbb{N} \mid L(M_z) \neq \emptyset\}\) is \(\Sigma^1_1\)-complete.

2. \(\{z \in \mathbb{N} \mid L(M_z) \text{ is infinite}\}\) is \(\Sigma^1_1\)-complete.

We now state our first new result.

Lemma 3.2 \(\{z \in \mathbb{N} \mid L(M_z) \text{ is countably infinite}\}\) is in the class \(D_2(\Sigma^1_1)\).

Proof. We first show that \(\{z \in \mathbb{N} \mid L(M_z) \text{ is countable}\}\) is in the class \(\Pi^1_1\). Notice that here “countable” means “finite or countably infinite.”

We know that an \(\omega\)-language \(L(M_z)\) accepted by a Turing machine \(M_z\) is a \(\Sigma^1_1\)-subset of \(\Sigma^\omega\). But it is known that a \(\Sigma^1_1\)-subset \(L\) of \(\Sigma^\omega\) is countable if and only if for every \(x \in L\) the singleton \(\{x\}\) is a \(\Delta^1_1\)-subset of \(\Sigma^\omega\), see [Mos80, page 243].

On the other hand the following result is proved in [HKL90, Theorem 3.3.1]. There exists a \(\Pi^1_1\)-set \(W \subseteq \mathbb{N}\) and a \(\Pi^1_1\)-set \(C \subseteq \mathbb{N} \times \Sigma^\omega\) such that, if we denote \(C_n = \{x \in \Sigma^\omega \mid (n, x) \in C\}\), then \(\{(n, \alpha) \in \mathbb{N} \times \Sigma^\omega \mid n \in W \text{ and } \alpha \notin C_n\}\) is a \(\Pi^1_1\)-subset of the product space \(\mathbb{N} \times \Sigma^\omega\) and the \(\Delta^1_1\)-subsets of \(\Sigma^\omega\) are the sets of the form \(C_n\) for \(n \in W\).

We can now first express \((\exists n \in W \ C_n = \{x\})\) by the sentence \(\phi(x)\):

\[\exists n [ n \in W \text{ and } (n, x) \in C \text{ and } \forall y \in \Sigma^\omega [(n \in W \text{ and } (n, y) \notin C) \text{ or } (y = x)]\]

But we know that \(C\) is a \(\Pi^1_1\)-set and that \(\{(n, \alpha) \in \mathbb{N} \times \Sigma^\omega \mid n \in W \text{ and } \alpha \notin C_n\}\) is a \(\Pi^1_1\)-subset of \(\mathbb{N} \times \Sigma^\omega\). Moreover the quantification \(\exists n\) in the above formula is a first-order quantification therefore the above formula \(\phi(x)\) is a \(\Pi^1_1\)-formula.
We can now express that \( L(M_z) \) is countable by the sentence \( \psi(z) : \)

\[
\forall x \in \Sigma^\omega \ [(x \notin L(M_z)) \text{ or } (\exists n \in W \ C_n = \{x\})]
\]

that is,

\[
\forall x \in \Sigma^\omega \ [(x \notin L(M_z)) \text{ or } \phi(x)]
\]

We know from [CC89] that \( x \notin L(M_z) \) is expressed by a \( \Pi^1_1 \)-formula. Thus the above formula \( \psi(z) \) is a \( \Pi^1_1 \)-formula. This proves that the set \( \{ z \in \mathbb{N} \mid L(M_z) \text{ is countable} \} \) is in the class \( \Pi^1_1 \).

On the other hand the set \( \{ z \in \mathbb{N} \mid L(M_z) \text{ is infinite} \} \) is in the class \( \Sigma^1_1 \). Finally the set \( \{ z \in \mathbb{N} \mid L(M_z) \text{ is countably infinite} \} \) is the intersection of a \( \Sigma^1_1 \)-set and of a \( \Pi^1_1 \)-set, i.e. it is in the class \( D_2(\Sigma^1_1) \). \( \square \)

We now give the exact complexity for this decision problem.

**Theorem 3.3** \( \{ z \in \mathbb{N} \mid L(M_z) \text{ is countably infinite} \} \) is \( D_2(\Sigma^1_1) \)-complete.

**Proof.** Recall that Castro and Cucker proved in [CC89] Proof of Proposition 3.1] that there is a computable injective function \( \varphi \) from \( \mathbb{N} \) into \( \mathbb{N} \) such that there are two cases:

**First case:** \( z \in E_1 \) and \( L(M_{\varphi(z)}) = \Sigma^\omega \).

**Second case:** \( z \in E_1^- \) and \( L(M_{\varphi(z)}) = \emptyset \).

We can easily define injective computable functions \( g \) and \( h \) from \( \mathbb{N} \) into \( \mathbb{N} \) such that for every integer \( z \in \mathbb{N} \) it holds that :

\[
L(M_{g(z)}) = L(M_z) \cup a^* \cdot b^\omega
\]

and

\[
L(M_{h(z)}) = L(M_z) \cap a^* \cdot b^\omega
\]

We can see that there are now two cases:

**First case:** In this case \( z \in E_1 \) and \( L(M_{\varphi(z)}) = \Sigma^\omega \). Thus \( L(M_{g \circ \varphi(z)}) = \Sigma^\omega \) is uncountable and \( L(M_{h \circ \varphi(z)}) = a^* \cdot b^\omega \) is countable.

**Second case:** In this case \( z \in E_1^- \) and \( L(M_{\varphi(z)}) = \emptyset \). Thus \( L(M_{g \circ \varphi(z)}) = a^* \cdot b^\omega \) is countable and \( L(M_{h \circ \varphi(z)}) = \emptyset \).

We define now the following simple operation over \( \omega \)-languages. For two \( \omega \)-words \( x, x' \in \Sigma^\omega \) the \( \omega \)-word \( x \oplus x' \) is defined by : for every integer \( n \geq 1 \)

\[
(x \oplus x')(2n-1) = x(n) \quad \text{and} \quad (x \oplus x')(2n) = x'(n).
\]

For two \( \omega \)-languages \( L, L' \subseteq \Sigma^\omega \), the \( \omega \)-language \( L \oplus L' \) is defined by \( L \oplus L' = \{ x \oplus x' \mid x \in L \text{ and } x' \in L' \} \).
It is easy to see that there is a computable injective function $\Phi$ from $\mathbb{N}^2$ into $\mathbb{N}$ such that for every integer $z, z' \in \mathbb{N}$ it holds that

$$L(\mathcal{M}_{\Phi(z,z')}) = L(\mathcal{M}_z) \oplus L(\mathcal{M}_{z'})$$

We want now to show that every subset of $\mathbb{N}$ in the class $D_2(\Sigma_1^1)$ is $1$-reducible to the set $\{z \in \mathbb{N} \mid L(\mathcal{M}_z) \text{ is countably infinite}\}$.

Let $E$ be a $D_2(\Sigma_1^1)$-subset of $\mathbb{N}$. Then there are some sets $A \subseteq \mathbb{N}$ and $B \subseteq \mathbb{N}$ such that $A$ is a $\Sigma_1^1$-set and $B$ is a $\Pi_1^1$-set and $E = A \cap B$. But the set $E_1$ is $\Sigma_1^1$-complete and the set $E_1^\perp$ is $\Pi_1^1$-complete. Thus there are some injective computable mappings $f_A$ and $f_B$ from $\mathbb{N}$ into $\mathbb{N}$ such that $A = f_A^{-1}(E_1)$ and $B = f_B^{-1}(E_1^\perp)$.

It is easy to see that there is an injective computable function $\Psi$ from $\mathbb{N}$ into $\mathbb{N}$ such that for every $z \in \mathbb{N}$ it holds that :

$$L(\mathcal{M}_{\Psi(z)}) = L(\mathcal{M}_{\Phi(h_{\varphi\psi}f_A(z), g_{\varphi\psi}f_B(z))}) = L(\mathcal{M}_{h_{\varphi\psi}f_A(z)}) \oplus L(\mathcal{M}_{g_{\varphi\psi}f_B(z)})$$

We next show that $\Psi$ is a reduction. We divide into cases.

**First case:** $z \in E = A \cap B$. Then $f_A(z) \in E_1$ and $L(\mathcal{M}_{h_{\varphi\psi}f_A(z)}) = a^* \cdot b^\omega$ is countably infinite. Moreover $f_B(z) \in E_1^\perp$ and $L(\mathcal{M}_{g_{\varphi\psi}f_B(z)}) = a^* \cdot b^\omega$ is countably infinite. Thus the $\omega$-language $L(\mathcal{M}_{\Psi(z)}) = L(\mathcal{M}_{h_{\varphi\psi}f_A(z)}) \oplus L(\mathcal{M}_{g_{\varphi\psi}f_B(z)})$ is also countably infinite.

**Second case:** $z \notin E = A \cap B$. Then either $z \notin A$ or $z \notin B$. Assume first that $z \notin A$. Then $f_A(z) \notin E_1$, i.e. $f_A(z) \in E_1^\perp$. Thus $L(M_{h_{\varphi\psi}f_A(z)}) = \emptyset$. Assume now that $z \notin B$, i.e. $f_B(z) \notin E_1$. Then $L(M_{g_{\varphi\psi}f_B(z)}) = \Sigma^\omega$. We can see that if either $z \notin A$ or $z \notin B$ the $\omega$-language $L(M_{\Psi(z)}) = L(M_{h_{\varphi\psi}f_A(z)}) \oplus L(M_{g_{\varphi\psi}f_B(z)})$ can not be countably infinite because it can only be either empty or uncountable.

Finally, using the reduction $\Psi$ we have proved that

$$E \leq_1 \{z \in \mathbb{N} \mid L(\mathcal{M}_z) \text{ is countably infinite}\}$$

so this latter set is $D_2(\Sigma_1^1)$-complete.

**Remark 3.4** Castro and Cucker noticed in [CC89] that the set $\{z \in \mathbb{N} \mid L(\mathcal{M}_z) \text{ is countably infinite}\}$ is in the class $\Sigma_2^1$ but they asked whether this set is $\Sigma_1^1$-complete. Our result shows that the answer is “no” because a $D_2(\Sigma_1^1)$-set is actually much less complex than a $\Sigma_1^1$-complete set.
Recall that an \( \omega \)-language accepted by a Turing machine \( M \) is a \( \Sigma_1^1 \)-subset of \( \Sigma^\omega \). Then it is well known that such a set is either countable or has the cardinal \( 2^{\aleph_0} \) of the continuum, see [Mos80]. Therefore an \( \omega \)-language accepted by a Turing machine has cardinal \( 2^{\aleph_0} \) iff it is not a countable set.

We can now state the following result.

**Theorem 3.5**

1. \( \{ z \in \mathbb{N} \mid L(M_z) \text{ is uncountable} \} \) is \( \Sigma_1^1 \)-complete.

2. \( \{ z \in \mathbb{N} \mid L(M_z) \text{ is countable} \} \) is \( \Pi_1^1 \)-complete.

**Proof.** We first prove item (1).

We have already seen that \( \{ z \in \mathbb{N} \mid L(M_z) \text{ is countable} \} \) is in the class \( \Pi_1^1 \).

Thus the set \( \{ z \in \mathbb{N} \mid L(M_z) \text{ is uncountable} \} \) is a \( \Sigma_1^1 \)-set.

To prove the completeness result we can use an already cited result of Castro and Cucker. There is a computable injective function \( \varphi \) from \( \mathbb{N} \) into \( \mathbb{N} \) for which one of the two following cases hold:

**First case:** \( z \in E_1 \) and \( L(M_{\varphi(z)}) = \Sigma^\omega \).

**Second case:** \( z \in E_1^- \) and \( L(M_{\varphi(z)}) = \emptyset \).

The reduction \( \varphi \) shows that :

\[ E_1 \leq_1 \{ z \in \mathbb{N} \mid L(M_z) \text{ is uncountable} \} \]

so this latter set is \( \Sigma_1^1 \)-complete.

Item (2) follows directly from Item (1).

**References**


