Expressive power versus decidability

Reijo Jaakkola
Tampere University, Finland

Abstract
In this note we prove that there exists no fragment of first-order logic which satisfies simultaneously the following requirements: a) it has a recursive syntax b) it is equi-expressive with first-order logic over finite models c) it has a decidable finite satisfiability problem d) it is effectively closed under conjunction. We also point out that there exists a fragment of first-order logic which satisfies requirements a), b) and c) simultaneously.

1 Introduction
It is well-known that both the general satisfiability problem and the finite satisfiability problem of first-order logic FO are undecidable [1, 5, 4]. These negative results directed the research towards syntactical fragments of FO, with the hope that expressive logics with good computational properties will be discovered.

All of the decidable fragments of FO that have been discovered so far are much weaker than FO with respect to expressive power. For instance, already

$$\forall x \forall y \forall z (R(x, y, z) \lor R(z, y, x))$$

is a sentence which is easily seen to be not, for example, expressible in two-variable logic $\mathsf{FO^2}$, unary negation fragment UNF or fluted logic FL (for definitions of these fragments and for an introduction to the topic of fragments of FO, see [3]). This motivates the following question: can there be a decidable fragment of FO which has the same expressive power as FO?

If we do not limit the fragments that we can use in any way, then the answer is quite trivially “yes”, since

$$\{\bot\} \cup \{\phi \in \mathsf{FO} \mid \phi \text{ is satisfiable}\}$$

is a decidable fragment of FO which has the same expressive power as FO. However, the above set does not have any effective syntax. Can this be avoided?

The answer is clearly “no”, since an easy consequence of the fact that the set of valid sentences of FO is computably enumerable is that if $\mathcal{L} \subseteq \mathsf{FO}$ is computable and it has the same expressive power as FO, then FO can be translated effectively into $\mathcal{L}$. In particular, the satisfiability problem of FO can be reduced effectively to that of $\mathcal{L}$ and hence there is no decidable fragment of FO that has the same expressive power as FO (if we restrict our attention to those fragments which have an effective syntax).

What about the case of finite models? By Trakhtenbrot’s theorem, the set of finitely valid sentences of FO is not computably enumerable, and so the argument
that we applied previously can not be applied. In fact it turns out that there are
computable sets \( L \subseteq \text{FO} \) which have a decidable finite satisfiability problem and
are equi-expressive with FO over finite models. An example of such a set is
\[ \{ \bot \} \cup \{ \varphi^n | \varphi \in \text{FO} \text{ and } \varphi \text{ has a model of size } n \}, \]
where \( \varphi^n := \varphi \land \cdots \land \varphi \) \( n \)-times.

Fragments such as the one mentioned above are not particularly natural, and
hence one might wonder how they could be excluded. The main purpose of this
note is to show that it suffices to assume that the fragments under question are
effectively closed under conjunction.

**Theorem 1.1.** There exists no computable set \( L \subseteq \text{FO} \) which has the same expres-
sive power as \( \text{FO} \) over finite models, is effectively closed under conjunction and has
a decidable finite satisfiability problem.

A very pleasant aspect of Theorem 1.1 is that it formalizes and verifies the
intuition that the decidability of a logical system is closely related to its expressive
power, an intuition which underlines most of the research around fragments of FO.

## 2 Preliminaries

We fix some reasonable enumeration \((M_x)_{x \in \{0, 1\}^*}\) of all Turing machines. Thus the
binary string \( x \) encodes the Turing machine \( M_x \).

We fix a computable set \( \mathcal{V} \) of relation symbols and use FO to denote the set of
all sentences of first-order logic over the vocabulary \( \mathcal{V} \). For our purposes, it would
suffice to assume that \( \mathcal{V} \) is finite (even a single binary relation would suffice).

Fragments of FO are essentially computable subsets \( L \subseteq \text{FO} \). Most of them are
closed under conjunction syntactically, but there are also those fragments which are
only closed under conjunction with respect to expressive power (for instance some
of the prefix fragments). This motivates the following definition.

**Definition 2.1.** Let \( L \subseteq \text{FO} \) be a computable set. We say that \( L \) is **effectively
closed under conjunction**, if there exists a computable mapping \( f : L \times L \rightarrow L \)
so that \( f(\varphi, \psi) \) is equivalent with \( \varphi \land \psi \).

**Word** is a finite structure \( \mathfrak{A} \) over a vocabulary consisting of unary relation
symbols and a single binary relation symbol \( < \) which satisfies the following two
requirements.

1. \(<^\mathfrak{A}\) is a linear ordering of \( A \).
2. For every \( a \in A \) there exists precisely one unary relation symbol \( P \) in the
   underlying vocabulary for which \( a \in P^\mathfrak{A} \).

The following well-known fact is proved in [2].

**Fact 2.2.** The satisfiability problem of FO over words is decidable.
3 Proof of the main result

Let $\mathcal{L} \subseteq \text{FO}$ be a computable set which has the same expressive power as FO over finite models and is effectively closed under conjunction. Our goal is to show that we can effectively reduce halting problem to the finite satisfiability problem of $\mathcal{L}$.

The chief technical obstacle that we need to bypass is that we cannot in general translate effectively FO-sentences to sentences of $\mathcal{L}$. However, by Fact 2.2, we know that we can effectively determine whether two sentences FO are equivalent over words, and hence any such FO-sentence $\varphi$ can be effectively translated to a sentence $\varphi' \in \mathcal{L}$ which is equivalent with $\varphi$ over words. Our general strategy is now that we will use these sentences to encode inputs to a universal Turing machine, which in turn will be encoded with a single fixed sentence of $\mathcal{L}$.

Now we will proceed with the formal proof. Let $U$ be a universal Turing machine, which when given as an input a binary string $x$, simulates the run of $M_x$ on empty input. Thus $U$ halts on $x$ if and only if $M_x$ halts on empty input. We will assume that the vocabulary of $U$ is $\{\square, 0, 1\}$ and that $U$ has a single one-way infinite memory tape. We will next describe sentences $\varphi_U$ and $\varphi_x$ which have the following properties.

1. $\varphi_U \land \varphi_x$ is finitely satisfiable iff $U$ halts on input $x$.

2. For every model $\mathfrak{A}$ of $\varphi_U$ the restriction of $\mathfrak{A}$ to $\{E, Z, O, <\}$ is a word.

We start with the sentence $\varphi_x$. Consider the vocabulary $\tau_0 = \{E, Z, O, <\}$, where $E, Z, O$ are unary while $<$ is binary. $\varphi_x$ will be the following sentence:

$$\exists x_1 \ldots \exists x_n [\text{first}(x_1) \land \bigwedge_{1 \leq i < n} \text{succ}(x_i, x_{i+1})$$

$$\land \bigwedge_{x_i = 0} \text{Z}(x_i) \land \bigwedge_{x_i = 1} \text{O}(x_i) \land \forall y (x_n < 1 \rightarrow E(y))]$$

Here $\text{first}(z)$ describes the fact that $z$ is the smallest element in $<$, while $\text{succ}(z, w)$ describes the fact that $w$ is the immediate successor of $z$ in $<$. Thus over words the sentence $\varphi_x$ expresses the fact that the underlying word is of the form $x\square \ldots \square$, where each $\square$ denotes a “blank” symbol.

Next we will consider the sentence $\varphi_U$. We will describe the computation of $U$ as a finite grid, where the $i$:th row describes a sufficiently large prefix of the memory tape of $U$ at stage $i$. In addition to $\tau_0$, the vocabulary of $\varphi_U$ will contain in addition at least three binary relation symbols $P_{\text{zero}}, P_{\text{one}}, P_{\text{empty}}$, which can be used to describe the content of the memory tape of $U$; and a binary relation symbol $<'$ which will be used to describe a second linear ordering on the models domain. Roughly speaking, in the finite grid the element at $(i, j)$ correspondence to a pair of elements in the domain where the first element is the $i$:th element in $<'$ while the second element is the $j$:th element in $<$.

Given this intuition, it is mostly routine to write down $\varphi_U$. The only non-standard part of $\varphi_U$ is the manner in which we encode the input received by $U$, since we want $\varphi_U$ to be independent of $x$. The following sentence

$$\exists x [\text{first}'(x) \land$$

$$\forall y ((\text{Z}(y) \rightarrow P_{\text{zero}}(x, y)) \land (\text{O}(y) \rightarrow P_{\text{one}}(x, y)) \land (\text{E}(y) \rightarrow P_{\text{empty}}(x, y)))]$$

3
where first′(x) describes that x is the smallest element in <′, will take care of this, since together with the sentence ϕx it encodes the fact that the bottom row in the finite grid – essentially the input tape of U – starts with the binary word x followed by a sequence of empty “squares”.

To complete the proof, we will describe how a sentence of L equivalent with ϕU ∧ ϕx can be obtained effectively from the input string x. First, since the sentence ϕU does not depend on the word x, we can hardcode an equivalent sentence ϕ′ U ∈ L into our procedure. Secondly, since the satisfiability problem of FO over words is decidable, we can effectively find a sentence of L which is equivalent with the sentence ϕx over words; let ϕ′ x ∈ L denote this sentence. Since L is effectively closed under conjunction, we can effectively compute a third sentence χx which is equivalent with ϕ′ U ∧ ϕ′ x.

4 Conclusions

In this note we have proved that there exists no computable subset of FO which is equi-expressive with FO over finite models, is effectively closed under conjunction and has a decidable finite satisfiability problem. This result demonstrates formally that the expressive power of a logic is intimately related to its decidability.

The proof presented here also neatly explains this connection between expressive power and decidability (or why it does not matter what syntax we use, as long as it satisfies some minimal requirements). Indeed, it all comes down to the fact that there exists a universal Turing machine, since as long as we are able to describe its behaviour, we just need to encode its inputs, which are very simple to describe.

Acknowledgements

The author wishes to thank Antti Kuusisto for extremely useful discussions.

References

[1] A. Church. A note on the entscheidungsproblem. J. Symb. Log., 1:40–41, 1936.
[2] L.J. Stockmeyer. The Complexity of Decision Problems in Automata Theory and Logic. Massachusetts Institute of Technology, Project MAC, 1974.
[3] Lidia Tendera. Finite model reasoning in expressive fragments of first-order logic. In M4M@ICLA, 2017.
[4] B. A. Trakhtenbrot. Impossibility of an algorithm for the decision problem on finite classes (in russian). Doklady Akademii Nauk SSSR, 70:569–572, 1950.
[5] A. Turing. On computable numbers, with an application to the entscheidungsproblem. Proceedings of The London Mathematical Society, 41:230–265, 1937.