Nico Bachmann
Ruben Hutter

Exercise 8.1

KEI PLAN WAS SIE MIT SET S MEINT...

(a) Let M_1 be a TM that is undefined on input 0 and M_2 be another TM that is defined on input 0. Let both TM's compute an unary function over $\mathbb{N}$. The Rice theorem tells us that it's impossible to write a general algorithm that decides whether a given TM has a property or not except if the property is always true or false. Since M_1 and M_2 are two TM's that behave differently, the property of 0 being undefined is not always true or false and therefore the language L is undecidable.

(b) Let M_1 be a TM that rejects every input and M_2 be another TM that accepts every input. Both TM's obviously behave differently and are therefore undecidable using the Rice theorem

(c) L is decidable because if it halts on an even number of steps for input 0, it always halts on an even number of steps. (KEI PLAN WIESO... CHATGPT)

(d) L is decidable because we can "simply" compute possible pairs of input and compare them with the output. (EBEFALLS CHATGPT. MACHT IRGENDWIE HALBWEGS SINN??

Exercise 8.2

Since there exist functions to transform a given type-0 grammar to a DTM and vice versa, we can look at the problem from a DTM perspective.
Let $S = \{L(M) | L(M) = \emptyset\}$ be the set of languages that contains all languages of TM's that have an empty language.

Let M_1 be a DTM with $L(M_1) = \emptyset$. For example M_1 could reject every input immediately. Then let M_2 be a DTM with $L(M_2) \neq \emptyset$. For example M_2 could accept a single string.

In this case, $L(M_1) \in S$ but $L(M_2) \notin S$ and we see that S is a non-trivial property of Turing Machine languages.

Now Rice theorem tells us that every non-trivial property of a language is undecidable and because there exist functions to transform from a DTM to a type-0 grammar, we can apply this result also for the type 0 grammars. Therefore $EMPTINESS$ is in fact undecidable.

Exercise 8.3

(a)

(b)

Nico Bachmann
Ruben Hutter

Exercise 8.4

(a) n^3 will dominate as n gets larger. Therefore the runtime of X is bound to a polynomial function and we can conclude that problem X is part of P.

(b) The runtime of $n^{log_2(n)}$ grows faster than any polynomial function because the exponent is not constant and grows with the size of n. We cannot conclude that X is in P for that reason.