Theory of Computer Science
Spring semester 2023
Exercise 8

Nico Bachmann
Ruben Hutter
Page 1 of 2

1/2 Exercise 8.1

(a) $S = \{ f \mid f \text{ is undefined for input } 0 \}$ ✓

(b) $S = \{ f \mid f \text{ is a total function and computable} \}$ ✗   rice not applicable

(c) There exists a TM that accepts the same language as a another TM but one halts on an even number of steps while the other on an uneven number of steps. ? is rice applicable?

(d) $S = \{ f \mid f \text{ computes the binary multiplication function mul: } \mathbb{N}_0^2 \to \mathbb{N}_0 \text{ with mul(x, y)} = x * y \}$ ✓

2/3 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 $S = \{L(M) \mid L(M) = \emptyset\}$ 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.

you somewhat got the idea, however languages and TMs are not functions, therefore the set S that you defined is not suitable for Rice (we need functions in there).

3/4 Exercise 8.3   Properties of TMs do not define a set of function   -1

(a) A non-deterministic algorithm for the Hitting Set problem is as follows:

Algorithm 1 Non-deterministic Algorithm for HittingSet (short version)

1: Choose a set H of size at most k non-deterministically.
2: **for** each set $S_i \in S$ **do**
3:     Check whether $S_i \cap H$ is non-empty.
4:     **if** $S_i \cap H$ is empty **then**
5:         Reject the instance.
6:     **end if**
7: **end for**
8: Accept the instance. ✓

The algorithm runs in polynomial time because each non-deterministic choice can be made in polynomial time. The size of the input is proportional to the size of the set U and the number of sets in S, so the runtime is polynomial in the size of the input. ✓

1

Theory of Computer Science
Spring semester 2023
Exercise 8

Nico Bachmann
Ruben Hutter
Page 2 of 2

The algorithm is correct because if there exists a hitting set of size at most k, then the algorithm will accept the instance. If there does not exist a hitting set of size at most k, then the algorithm will reject the instance for any choice of H of size at most k. Therefore, the algorithm correctly decides the decision version of the Hitting Set problem. ✓

Algorithm 2 Deterministic Algorithm for HittingSet (short version) xD

(b)
1: **for** each $H' \in$ all possible subsets of U with $|H'| \leq k$ **do**
2:     Check if $S_i \cap H' \neq \emptyset$ for all $Si \in S$.
3:         **return** True
4: **end for**
5: **return** False ✓

Exercise 8.4

1/1

(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 X is a P problem ✓

(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.
ok