Introduction to computability
Tutorial 10

Uncomputability and Complexity

04 December 2018

Some undecidable languages

universal language: $UL = \{\langle M, w\rangle \mid M \text{ accepts } w\}$

$\overline{UL} = \{\langle M, w\rangle \mid M \text{ rejects or cycles on } w\}$

halting problem: $H = \{\langle M, w\rangle \mid M \text{ stops on } w\}$

empty-word halting problem: $\{M \mid M \text{ stops on } \varepsilon\}$

existential halting problem: $\{M \mid (\exists w)\ M \text{ stops on } w\}$

universal halting problem: $\{M \mid (\forall w)\ M \text{ stops on } w\}$

empty accepted language: $\{M \mid L(M) = \varnothing\}$

recursive accepted language: $\{M \mid L(M) \in R\}$

undecidable accepted language: $\{M \mid L(M) \notin R\}$

1. Let L be a language such that $L \in RE$. Show that the problem that consists of determining if L is a finite language is undecidable.

1. Let L be a language such that $L \in RE$. Show that the problem that consists of determining if L is a finite language is undecidable.

2. Let M be a Turing machine and t one of its transitions. Show that it is impossible to determine algorithmically that the transition t is used in the execution of M on the empty word.

1. Let L be a language such that $L \in RE$. Show that the problem that consists of determining if L is a finite language is undecidable.

2. Let M be a Turing machine and t one of its transitions. Show that it is impossible to determine algorithmically that the transition t is used in the execution of M on the empty word.

3. Let M_1 and M_2 be two Turing machines. Show that the problem that consists of determining if the execution of M_1 on the empty word needs (strictly) fewer steps than the execution of M_2 on the empty word is undecidable.

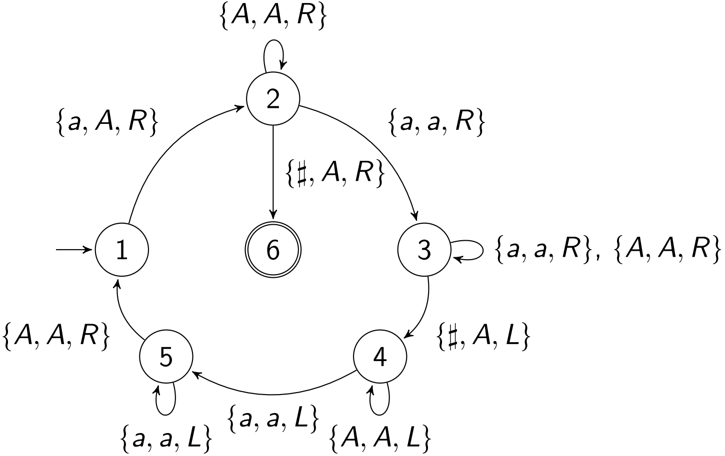
Complexity

The class P is the class of languages *decided* by a *polynomial deterministic* Turing Machine.

The class NP is the class of languages that are *accepted* by a *polynomial nondeterministic* Turing Machine.

The class NPC is the class of languages that are in NP and that are "hardest to decide".

4. What is the time complexity of the Turing machine below where the entry alphabet is $\Sigma = \{a\}$?



5.

- Are the regular languages in P?
- Are the context-free languages in P?

5.

- Are the regular languages in P?
- Are the context-free languages in P?

6. Show in a diagram the inclusion relations between P, NP, R, RE, the regular languages and the context-free languages.

5.

- Are the regular languages in P?
- Are the context-free languages in P?

6. Show in a diagram the inclusion relations between P, NP, R, RE, the regular languages and the context-free languages.

7. Give a deterministic algorithm for the Hamilton Circuit problem and estimate its complexity.

5.

- Are the regular languages in P?
- Are the context-free languages in P?

6. Show in a diagram the inclusion relations between P, NP, R, RE, the regular languages and the context-free languages.

7. Give a deterministic algorithm for the Hamilton Circuit problem and estimate its complexity.

8. Let L_1 and L_2 be two languages that belong to the same complexity class, P, NP or NPC. What is the complexity class of the languages $L_1 \cap L_2$ and $L_1 \cup L_2$?

Bonus Exercise 11

Give a deterministic algorithm to solve the SAT problem. What is its complexity?