Theory of Computation

Author: Matteo Ghilardini

Assignment 2

1. Let L be a decidable language. Prove that its complement L' is decidable.

Knowing that language L is decidable, we know that there exists a Turing Machine M that: - accepts if word $w \in L$ - rejects if word $w \notin L$ - is a decider (i.e. always terminates by accepting or rejecting any input)

We know that the complement L' of a language L is defined as the set of strings $w \notin L$.

That said, it is clear that if M is a decider for L, we can construct a Turing Machine M' that is decider for L' using M as a subcomponent since M' is defined as: - accepts if M rejects, i.e. accepts if word $w \notin L = w \in L'$ - rejects if M accepts, i.e. rejects if word $w \in L = w \notin L'$ - is a decider since M is itself a decider (if M ends in **accept**, M' will end in **reject** and vice versa)

This shows that if L is a decidable language, its complement L' will also be decidable since there exists M' which is a decider.

2. Let $ALL_{DFA} = \{\langle A\rangle | A$ is a DFA and $L(A) = \sum^*\}$, that is, the set of all the encodings of a DFA such that the language recognized by the automaton is the set of all words (over the alphabet $\sum$). Show that ALL_{DFA} is decidable.

To prove that ALL_{DFA} is decidable, we need to construct a Turing Machine M that is a decider (i.e. always terminates, accepting or rejecting).

We can therefore construct M as a turing machine that receives $\langle A\rangle$ (that is a DFA as defined by the language) as input and proceeds as follows: - Mark the initial state of A - As long as there are unmarked states, repeat: - Mark the states connected to the marked states (according to BFS or DFS) - If among the marked states, there are *non-accepting* states, then **reject** - If all the marked states are *accept*, then **accept**

So M will give the following output: - if $\langle A\rangle \notin ALL_{DFA}$ (i.e., all reachable states are *accepting*) => **accept** - if $c\langle A\rangle \notin ALL_{DFA}$ (i.e., **not** all reachable states are *accepting*) => **reject**

The key idea is that since a DFA has a finite number of states to control, the Turing Machine M will always terminate, i.e., it is a decider.