# CS 381 - A9

Martin Mueller

Due: May $11^{th}$, 2020

Let $A$ be be the set of all encodings $\langle M \rangle$, $M$ is a DFA, such that, for all $k \in \mathbb{N}$, there exists $m \in \mathbb{N}$, such that, $m \geq k$ and $M$ accepts a string of length $m$. Prove that $A$ is decidable. You may assume that your decider implicitly verifies the format of the input, but all other details must be specified and justified.

First, we will need a decider that can determine if a DFA recognizes the empty language. To do this, we define the language for which we want a decider $E_{\text{DFA}}$ such that $E_{\text{DFA}} = \{\langle A \rangle \mid A \text{ is a DFA and } L(A) = \emptyset\}$. Next, we know that a DFA accepts some string if and only if reaching an accept state from the start state by traveling along the DFA's transitions is possible. To test this, we can design a TM $T_1$ that marks the states which it has visited in order to determine if the DFA accepts a given string.

$T_1 =$ "On input $\langle A \rangle$ where $A$ is a DFA:

1. Mark the start state of $A$.
2. Repeat until no new states get marked:
   (a) Mark any state that has a transition going into it from any state that has already been marked.
3. If none of the accept states are marked, *accept*; otherwise, *reject*."

From the problem statement, we can now more compactly define the language of the TM we must construct to be $L$ such that $L = \{\langle M \rangle \mid M \text{ is a DFA and } L(M) \text{ is a countably infinite language}\}$. We now can construct a TM $T_2$ that decides $L$:

$T_2 =$ "On input $\langle M \rangle$, where $M$ is a DFA:

1. Let $x$ be the number of states of $M$.
2. Construct a DFA $A$ that accepts all strings of length $x$ or more.
3. Construct a DFA $B$ such that $L(B) = L(M) \cap L(A)$.
4. Determine if $L(B) = \emptyset$ using the decider $T_1$.
5. If $T_1$ accepts, *reject*; if $T_1$ rejects, *accept*."

A DFA that accepts infinitely many strings must accept strings of an arbitrary length. $T_2$ accepts these kinds of DFAs. In addition, if $T_2$ accepts a DFA, the DFA accepts some string of length $x$ or more, where $x$ is the number of states in the DFA. We can also use the necessary conditions that the pumping lemma for regular languages provides to generate infinitely many strings that are accepted by $M$.