Lecture Tue 4/21. Languages and Decision Problems

Vanilla $L$: \textbf{INSTANCE:} A string $x \in \Sigma^*$. \textbf{Name of the problem.}
$L \subseteq \Sigma^*$ \textbf{QUESTION:} Is $x \in L$? \textbf{Just "L".}

For any problem $P$, the language $L_P$ of the problem is the set of allowable \textbf{INSTANCES} for which the answer to the \textbf{QUESTION} is YES.

$\text{NE}_{\text{DFA}}$: \textbf{INST:} A DFA $M$ \textbf{Guess:} Is there a string $x$ that $M$ accepts? \textbf{E.g.} \iff $L(M) \neq \emptyset$? Instance $\text{APC}$ is "Given one machine": formally, a string $e(M)$ encoding a DFA.

Language: $L_{\text{NE}_{\text{DFA}}} = \{ \text{DFAs } M : L(M) \neq \emptyset \}$ \textbf{Expresses:} a language over the domain of DFAs

Also called $\text{NE}_{\text{DFA}}$ by itself.

(For Emphasis) $E_{\text{DFA}}$: \textbf{INST:} A DFA $M$ \textbf{Guess:} Is $L(M) = \emptyset$?

Within the domain of DFAs, $L_{\text{NE}_{\text{DFA}}}$ is the complement of $L_{\text{NE}_{\text{DFA}}}$.
Within $\Sigma^*$, it's \textbf{almost} a complement: $L_{\text{NE}_{\text{DFA}}} = (\Sigma^\ast \setminus L_{\text{NE}_{\text{DFA}}}) \setminus \{ \gamma \in \Sigma^* : \gamma \text{ does not encode a DFA }\}$.

All DFA: \textbf{INST:} A DFA $M = (Q, \Sigma, \delta, s, F)$ \textbf{Guess:} Is $L(M) = \Sigma^*$?

Note that if $M' = (Q, \Sigma, \delta, s, \mathcal{F} \setminus F)$ is the complementary DFA: $L(M') = \complement L(M)$, then $L(M) = \emptyset \iff L(M') = \Sigma^*$. So AllDFA for $M$ is equivalent to $E_{\text{DFA}}$ for $M'$.

And we can solve $E_{\text{DFA}}$ for $M'$ by solving $\text{NE}_{\text{DFA}}$ for $M$ and inverting the final yes/no answer.

\begin{itemize}
  \item The acceptance states of the given DFA $M$. \textbf{Note that we invert both.}
  \item Our answer to the decision problem $\text{NE}_{\text{DFA}}$
\end{itemize}
How do we solve the NEDFA problem? Given a DFA M...

Algorithm: Breadth-First Search. Linear

Kind-of runs in Order: $O(n)$ time.

By the Church-Turing Thesis, there is a deterministic Turing machine $T$ such that $L(T) = L_{NEDFA}$ and $T$ always halts. Hence the NEDFA problem and its language, are called decidable.

Formal Theorem: For any decidable language $L = \Sigma^*$, $\overline{L}$ is decidable too.

Proof: By definition, there is a DTM $T$ so $L(T) = L$ and $T$ halts for all inputs. Using the facts:

- We have a DFA $\epsilon$ with $q_0$, $q_f$, $T' = T$.
- By this form and $T$ being total, on any $x$, the computation emerges either at $q_f$ or at $q_0$.

Then $T'$ also is total, and since there is no non-halting possibility, $L(T') = \overline{L(T)} = \overline{L}$. Thus $\overline{L}$ is decidable.

We don't need to care what chars are on this path — whatever string is formed along the path is an $x$ that $M$ accepts.
Then what about \textsc{AllDFA}? In the text's step-by-step terms:

Given: a DFA $M$ for which we want to decide if $L(M) = \Sigma^*$?

1. Complement $M$ to the complemented DFA $M' = (Q, \Sigma, \delta, s, \Sigma \setminus F)$ of $M$.
2. Run algorithm \textsc{Dfs} solving \textsc{N12DFA} on $M'$.
3. If \textsc{Dfs} answers yes, we answer no.
   If \textsc{Dfs} answers no, we answer yes: $L(M') = \emptyset$, so $L(M) = \Sigma^*$.

All three steps always terminate and are correct, so \textsc{AllDFA} is decidable.

How about \textsc{AllNFA}? Given an NFA $N$, is $L(N) = \Sigma^*$?

Steps

\begin{itemize}
  \item \textbf{Issue:} $N' = (Q, \Sigma, \delta, s, F)$ in general does not complement the language.
  \item Convert $N$ into an equivalent DFA $M$. (\textit{Can take exponential time}.)
  \item Thus $L(N) = \Sigma^* \iff L(M) = \Sigma^*$: the answer to \textsc{AllNFA} for $M$ is yes.
\end{itemize}

1.2.3. same as above: we have reduced \textsc{AllNFA} to \textsc{AllDFA} computably. Not efficiently, but computably is good enough to show that \textsc{AllNFA} is decidable.

\textbf{NCFG}:

\textbf{Instance}: A CFG $G$

\textbf{Question}: Is $L(G) \neq \emptyset$? i.e. is there some terminal string $x$ such that $S \Rightarrow^* x$?

\textbf{EPS}:

\textbf{Instance}: A CFG $G$ (or an ASCII encoding $\mathcal{E}(G)$ of $G$)

\textbf{Query}: Is $\epsilon \in L(G)$, i.e. does $S \Rightarrow^* \epsilon$? Yes iff \textsc{Sendhilable}.

The first part of the Chomsky NF algorithm decides \textsc{EPS CFG}.

Now, to decide \textsc{NCFG}, given $G$, make $G'$ by changing all terminals to $\epsilon$.

\textsc{NCFG} is also decidable. How about \textsc{AllCFG}? FACT: Undecidable!