# Lecture Tue 4/21. Languages and Decision Problems



"Vanilla L": $L \subseteq \Sigma^*$ — INSTANCE: A string $x \in \Sigma^*$ — QUESTION: Is $x \in L$?

Name of the problem: "Just L"

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

$NE_{DFA}$: Abbreviates "Non-Emptiness Problem For DFAs"

INST: A DFA M

QUES: Is there a string x that M accepts? I.e. Is $L(M) \neq \emptyset$?

Instance type is "Given one machine" → formally, a string e(M) encoding a DFA.

Language: $L_{NE_{DFA}} = \{$DFAs $M : L(M) \neq \emptyset\}$ — Language over the domain of DFAs

$\equiv \{e(M) : M$ is a DFA and $L(M) \neq \emptyset\}$ — Expressly a language over ASCII then $\subseteq \{0,1\}^*$

Also called $NE_{DFA}$ by itself.

(E for Emptiness) $E_{DFA}$: INST: A DFA M (or its encoding e(M))

QUES: Is $L(M) = \emptyset$?

Within the domain of DFAs, $L_{E_{DFA}}$ is the complement of $L_{NE_{DFA}}$.

Within $\Sigma^*$, it's "almost a complement": $L_{E_{DFA}} = \left(\Sigma^* \setminus L_{NE_{DFA}}\right) \setminus \{y \in \Sigma^* : y$ does not encode a DFA$\}$.

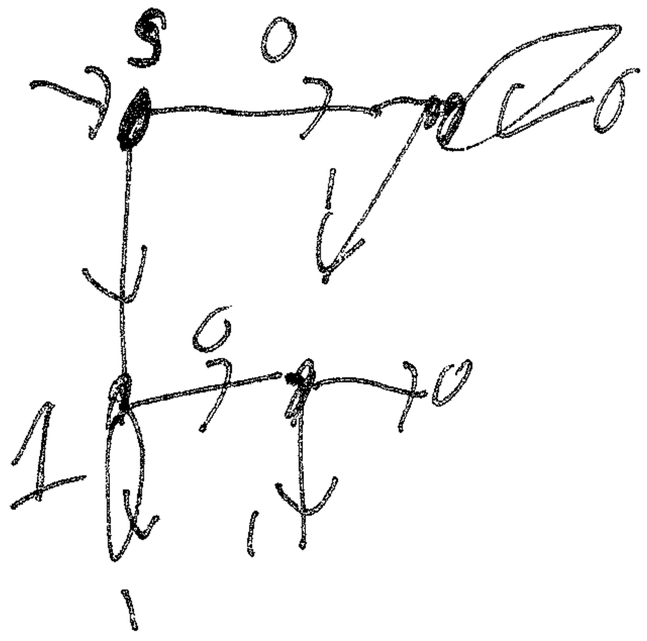
$ALL_{DFA}$: INST: A DFA $M = (Q, \Sigma, \delta, s, F)$

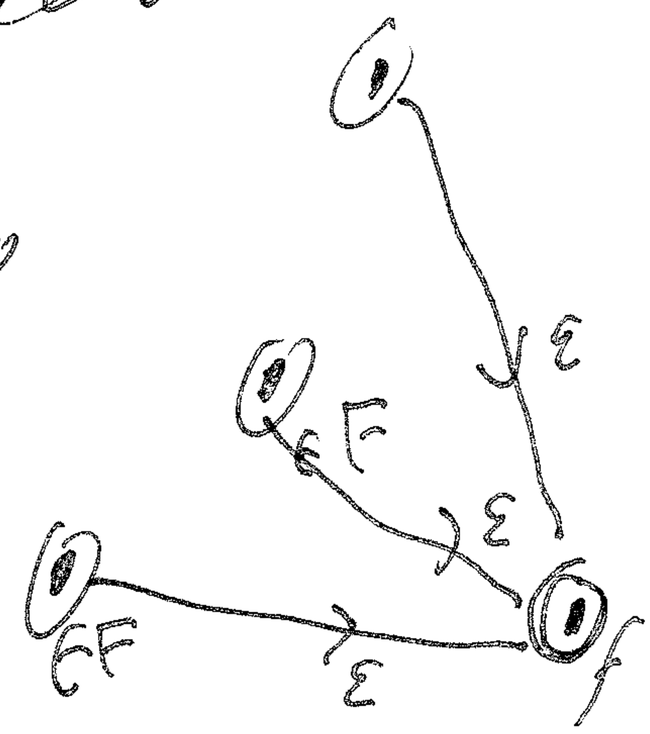
QUES: Is $L(M) = \Sigma^*$?

Note that if $M' = (Q, \Sigma, \delta, s, Q \setminus F)$ is the complementary DFA: $L(M') = \sim L(M)$, then $L(M) = \Sigma^* \iff L(M') = \emptyset$. So $ALL_{DFA}$ for M is equivalent to $E_{DFA}$ for M'. And we can solve $E_{DFA}$ for M' by solving $NE_{DFA}$ for M and inverting the final yes/no answer.

Note that we invert both
- The acc and rejecting states of the given DFA M
- Our answer to the decision problem $NE_{DFA}$

How do we solve the $NE_{DFA}$ problem? Given a DFA $M$... (2)

$M =$ 


Idea: $L(M) \neq \emptyset \Leftrightarrow$ there is some string $x \in \Sigma^*$ that $M$ accepts $\Leftrightarrow$ the graph of $M$ has a path from $s$ to $f$ (or to any final state)

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.

Algorithm: Breadth-First Search. Linear

("Kind-of" runs in Order-$|E(G)|$ time).

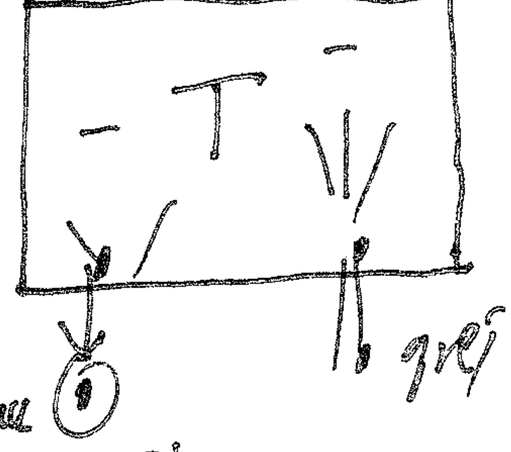
Unique Sink

∴ By the Programmer's Version of the Church-Turing Thesis, there is a deterministic Turing machine $T$ such that $L(T) = L_{NE_{DFA}}$ and $T$ always halts.

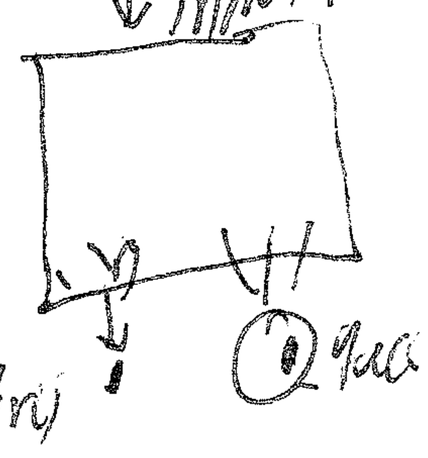
Hence the $NE_{DFA}$ problem, and its language, are called decidable.

∴ $E_{DFA}$ is decidable too — just invert the yes/no answer.

Formal Theorem: For any decidable language $L \subseteq \Sigma^*$, $\sim L$ is decidable too.

Proof: By definition, there is a DTM $T$ s.t. $L(T) = L$ and $T$ halts for all inputs. Using the text's nice form with $q_{acc}$, $q_{rej}$, $T =$ 


By this form and $T$ being total, on any $x$, the computation emerges either at $q_{acc}$ or at $q_{rej}$.

Now build $T'$ identically except for swapping $q_{acc}$ and $q_{rej}$ $T' =$ 


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

(i.e. $T$ is total)

Then what about $ALL_{DFA}$? In the text, step-by-step terms:

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

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

All three steps always terminate and are correct, so $ALL_{DFA}$ is decidable.

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

Steps

Issue: $N' = (Q, \Sigma, \delta, s, Q \setminus F)$ in general does not complement the language.

0. Convert $N$ into an equivalent DFA $M$. {Can Take Exponential Time.
   Thus $L(N) = \Sigma^* \iff L(M) = \Sigma^*$ the answer to $ALL_{DFA}$ for $M$ is yes.
1., 2. 3. Same as above: we have reduced $ALL_{NFA}$ to $ALL_{DFA}$ computably. Not efficiently, but computably is good enough to show that $ALL_{NFA}$ is decidable.

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

"$EPS_{CFG}$": INST: A CFG $G$ (or: an Ascii encoding $\langle G \rangle$ of $G$)
QUES: Is $\varepsilon \in L(G)$, ie does $S \Rightarrow^* \varepsilon$? Yes iff $S \in$ NULLABLE.

∴ The first part of the Chomsky NF algorithm decides $EPS_{CFG}$.

Now, to decide $NE_{CFG}$, given $G$, make $G'$ by changing all terminals to $\varepsilon$'s.

∴ $NE_{CFG}$ is also decidable. How about $ALL_{CFG}$? FACT. Undecidable!