CSE 396   Lecture Thu 4/18   Spring 2019

Top Hat 3787

Decision Problems About Computational Devices.

A_{DFA} – Acceptance Problem for DFAs.

INSTANCE: A DFA $M = (Q, \Sigma, \delta, s, F)$ and an argument $x \in \Sigma^*$

QUESTION: Does M accept x?

The language of the problem is the set of instances for which the answer is 'yes'. Often the same name of the problem is used for the language.

$A_{DFA} = \{\underbrace{\langle M, x \rangle}: M$ accepts x, i.e. $x \in L(M)\}$. Could write "$L_{A_{DFA}}$" but that's awkward.

some string encoding of M and x, possibly over ASCII. (don't care).

Fact: The A_{DFA} problem / language is decidable. To prove, we can sketch a decider in pseudocode.

Decider: On input $w = \langle M, x \rangle$,

1. Run the Turing Kit with M loaded and x on the tape.
2. It will halt since M is a DFA.
3. Hence the simulation will either accept x, in which case your routine accepts $\langle M, x \rangle$, else it will reject, so your routine halts and rejects.

I.e. "Just run it".

E_{DFA}: INST: Just a DFA M (no "x") QUES: Is $L(M) = \emptyset$?

Language $= \{\langle M \rangle : M$ is a DFA and $L(M) = \emptyset\}$

NE_{DFA}: INST: A DFA M QUES: Is $L(M) \neq \emptyset$?

As a language, this is $\{$ DFAs $\langle M \rangle : L(M) \neq \emptyset\} \approx \widetilde{E}_{DFA}$

not technically the complement since strings that are not valid $\langle M \rangle$ belong to neither

Because the complement of a decidable language is decidable
(and because in every prog. language except C++ we can decide whether
a program code will compile, full template metaprogramming!

to decide E_{DFA} it suffices to give a decider for NE_{DFA}.

FACT: $L(M) \neq \emptyset \iff$ there is a path in the graph of M
from start to some accepting state.

NB: If $F = \{s\}$ and no other strings besides ε get accepted we still have $L(M) = \{\varepsilon\} \neq \emptyset$, so $\langle M \rangle$ is in the NE_{DFA} language.
The path in this case has 0 steps. The computation on ε is just (s).

We can enumerate all states reachable from s by paths using BFS
Was already exemplified for the NFA-to-DFA construction. breadth-first search
In this case, however, the n possible states of M that might be seen
is known in advance. So BFS runs in $O(n^2)$ time, no "exponential explosion".
Thus BFS gives an (efficient) decider for NE_{DFA}, hence also E_{DFA}.

How about $NE_{NFA} = \{\text{NFAs } N : L(N) \neq \emptyset\}$?

The same FACT holds for NFAs N, so we can use the same BFS proc which has the same efficiency.
Thus E_{NFA} is (efficiently) decidable too.

How about $ALL_{DFA} = \{\text{DFAs } M : L(M) = \Sigma^*\}$? We can reduce to the already solved E_{DFA} problem by complementing M to M' s.t. $L(M') = \sim L(M)$.

Decider: 1. Convert M to M' (efficient: $F \mapsto F' = Q \setminus F$)
2. Run our E_{DFA} decider on $\langle M' \rangle$, and accept iff it accepts.
correct since $L(M) = \Sigma^* \iff L(M') = \emptyset$

How about $ALL_{NFA} = \{\text{NFAs } N : L(N) = \Sigma^*\}$?
Decidable but likely not efficiently. The problem is NP-hard (Ch. 7)

$EQ_{DFA} = \{\langle M_1, M_2\rangle : M_1$ and M_2 are DFAs and $L(M_1) = L(M_2)\}$.

Key Idea: $L(M_1) = L(M_2) \iff \underline{L(M_1) \,\Delta\, L(M_2)} = \emptyset$.

Decider: $\boxed{\text{1. Build } M_3 \text{ as at right.}}$
2. Run the E_{DFA} decider on $\langle M_3\rangle$
3. Accept $\langle M_1, M_2\rangle$ iff step 2 accepts.

- Get a DFA M_3 s.t. $L(M_3) = L(M_1) \,\Delta\, L(M_2)$
- Do Cartesian Product construction but with $F_3 = \{(p, q) : p \in F_1$ XOR $q \in F_2\}$.

NEQ_{DFA} is (essentially) the complement of EQ_{DFA}, hence decidable.

But EQ_{NFA} is reducible from ALL_{NFA}, $EQ_{NFA} = \{\langle N_1, N_2\rangle$: N_1 and N_2 are NFAs and $L(N_1) = L(N_2)\}$

by building $N_1 = N$, $N_2 =$ 

which is a NFA.

Nevertheless, is decidable by converting N_1 and N_2 into equivalent DFAs M_1 and M_2, then using the EQ_{DFA} decider. Just not efficient in conversi

E_{CFG} INST: A context free grammar $G = (V, \Sigma, R, S)$
QUES: Is $L(G) = \emptyset$?

Essential Complement $NE_{CFG} = \{\langle G\rangle : G$ is a CFG and $L(G) \neq \emptyset\}$.

Is NE_{CFG} decidable? Is it efficiently decidable?
By a loop "like" BFS?

Similar problem

ε-CFG: INST: G
QUES: Is $\varepsilon \in L(G)$, ie does $S \Rightarrow^* \varepsilon$?