Decidable and Undecidable Languages/Decision Problems

"G&J Format"

<table>
<thead>
<tr>
<th>Name &amp; Problem</th>
<th>Non-Emptyness for DFXs</th>
</tr>
</thead>
<tbody>
<tr>
<td>INSTANCE</td>
<td>INST: A DFA $M = (Q, \Sigma, \delta, F)$ encoded as a (binary) string in some natural way</td>
</tr>
<tr>
<td>QUESTION</td>
<td>QUES: Is $L(M) \neq \emptyset$? i.e., is there a string that $M$ accepts?</td>
</tr>
</tbody>
</table>

The language of a problem is always the set of legal instances for which the answer is yes.

Sipser Naming Scheme

[Type & Problem] (subscripted by objects it applies to)

Theorem: $\text{NE} \subseteq \text{NFA}$ is decidable in polynomial time.

Proof: Here is a pseudocode sketch of a polynomial decision procedure.

Helpful picture:

$M = \text{BFS runs in } \Theta(|V|)$ time.

Now how about the problem $\text{NE} \cap \text{NFA}$?

Same alg, same running time.
How about \( \text{ALL} \text{DFA} = \{ \text{DFA}, M : (Q, \Sigma, \delta, s_0, F) : \text{L}(M) = \Sigma^* \} \)?

Is this in \( P \)?

- **Fact:** \( \text{L}(M) = \Sigma^* \) \( \Rightarrow \text{L}(M) = \emptyset \).

How about \( \text{ALL} \text{NFA} = \{ \text{NFA}, N : \text{L}(N) = \Sigma^* \} \)?

**What if** \( N = \) \( \text{converting} \) \( N \) \( \text{as a DFA} \) \( M \) \( \text{and using ALLDFA on} \) \( M \) \( \text{is a decision procedure, just not in polynomial time.} \)

Fact: \( \text{ALLNFA is NP-hard! Hence unlikely to be in P.} \) \( \text{ALLNFA \ \not\in \ P \ \text{not even EXP.}} \)

How about \( \text{NE} \text{DFA} \):

**Inst.** Two DFAs \( M_1, M_2 \)

**Question:** is \( \text{L}(M_1) \cap \text{L}(M_2) = \emptyset ? \)

**Alg:** Build \( M_3 = \text{the Cartesian product of} \ M_1 \ \text{and} \ M_2 \) with \( F_3 = \emptyset / (q_1, q_2) : p \in F_1 \ \land \ q \in F_2 \) "the code".

Then \( \text{L}(M_3) = \text{L}(M_1) \cap \text{L}(M_2) \), so answer is \( \text{yes} \leftrightarrow <M_3> \in \text{NE} \text{DFA} \).

We can build \( M_3 \) in time \( \text{N} \times \text{N} \) \( (M, \text{has} n, \text{states}, M_1, \text{has} n_1) \) \( \times \) \( \text{N} \times \text{N} \) \( (M, \text{has} n, \text{states}, M_2, \text{has} n_2) \) then run BFS on \( M_3 \), \( \text{N} \times \text{N} \) time. \( \text{So it is in P.} \)

How about \( \text{NE} \text{NFA?} \)

**In P:** \( \text{Fix} \ N_1 \ \text{to acc} \ \Sigma^* \) \( N \in \text{NFA} \) \( \Rightarrow \text{UNATURAL} \) then \( \text{L}(N_1) \triangle \text{L}(N_2) \) \( \not\in \emptyset \).

How about \( \text{NE} \oplus \text{DFA} \)?

**In P:** Build \( M_3 \) using XOR, run BFS. Fix \( N_1 \) to acc \( \Sigma^* \) \( N \in \text{NFA} \) \( \Rightarrow \text{UNATURAL} \) then \( \text{L}(N_1) \triangle \text{L}(N_2) \) \( \not\in \emptyset \).

How about \( \text{NE} \oplus \text{NFA?} \) (NP-hard!)

- **In P:** Build \( M_3 \) using XOR, run BFS.