بسم الله الرحمن الرحیم

Decidability

Ali Shakiba
ali.shakiba@vru.ac.ir
Vali-e-Asr University of Rafsanjan

What we are going to discuss?

- Decidable problems concerning
  - Regular languages
  - Context-free languages
- Undecidability
  - Diagonalization method
  - An unrecognizable language

“exploring the limits of algorithmic solvability”

Why?

knowing that a problem is algorithmically unsolvable

↕

to solve, we need to simplify it

$$A_{DFA} = \{\langle B, w\rangle \mid B \text{ is a DFA that accepts } w\}$$

Decidable

Theorem 4.1

$M =$ "On input $\langle B, w\rangle$, where B is a DFA and w is a string:

1. Simulate B on input w.
2. If the simulation ends in an accept state, *accept*. If it ends in a nonaccepting state, *reject*."

$$A_{NFA} = \{\langle B, w\rangle | B \text{ is an NFA that accepts } w\}$$

Decidable

Theorem 4.2

$N =$ "On input $\langle B, w\rangle$, where B is an NFA and w is a string:

1. Convert NFA B to an equivalent DFA C, using the procedure for this conversion given in Theorem 1.39.
2. Run TM M from Theorem 4.1 on input $\langle C, w\rangle$.
3. If M accepts, *accept*; otherwise, *reject*."

$A_{REX} = \{\langle R, w\rangle | R \text{ is a regular expression that generates } w\}$

Decidable

Theorem 4.3

$P =$ "On input $\langle R, w\rangle$, where R is a regular expression and w is a string:

1. Convert regular expression R to an equivalent NFA A by using the procedure for this conversion given in Theorem 1.54.
2. Run TM N on input $\langle A, w\rangle$.
3. If N accepts, *accept*; if N rejects, *reject*."

$$E_{DFA} = \{\langle B, w\rangle | B \text{ is a DFA and } L(B) = \emptyset\}$$

Theorem 4.4

T = "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:
3. Mark any state that has a transition coming into it from any state that is already marked.
4. If no accept state is marked, *accept*; otherwise, *reject*."

$$EQ_{DFA} = \{\langle A, B\rangle | A \text{ and } B \text{ are DFAs and } L(A) = L(B)\}$$

Theorem 4.5

$$L(C) = \left(L(A) \cap \overline{L(B)}\right) \cup \left(\overline{L(A)} \cap L(B)\right)$$

F = "On input $\langle A, B\rangle$, where A and B are DFAs:

1. Construct DFA C as described.
2. Run TM T from Theorem 4.4 on input $\langle C\rangle$.
3. If T accepts, *accept*. If T rejects, *reject*."

$$A_{CFG} = \{\langle G, w \rangle | G \text{ is a CFG that generates string } w\}$$

Decidable

Theorem 4.7

$S =$ "On input $\langle G, w \rangle$, where G is a CFG and w is a string:

1. Convert G to an equivalent grammar in Chomsky normal form.
2. List all derivations with $2n - 1$ steps, where n is the length of w; except if $n = 0$, then instead list all derivations with one step.
3. If any of these derivations generate w, *accept*; if not, *reject*."

$$E_{CFG} = \{\langle G \rangle | G \text{ is a } CFG \text{ and } L(G) = \emptyset\}$$

Decidable

Theorem 4.8

$R =$ "On input $\langle G \rangle$, where G is a CFG:

1. Mark all terminal symbols in G.

2. Repeat until no new variables get marked:

3. Mark any variable A where G has a rule $A \rightarrow U_1 U_2 \cdots U_k$ and each symbol $U_1, \ldots, U_k$ has already been marked.

4. If the start variable is not marked, *accept*; otherwise, *reject*."

$A = L(G)$ for some CFG G

Decidable

Theorem 4.9

$M_G =$ "On input w:

1. Run TM S on input $\langle G, w \rangle$.
2. If this machine accepts, *accept*; if it rejects, *reject*."

$$EQ_{CFG} = \{\langle G, H\rangle | G \text{ and } H \text{ are CFGs and } L(G) = L(H)\}$$

We'll do it later in Chapter 5, by reducing it to HALT!

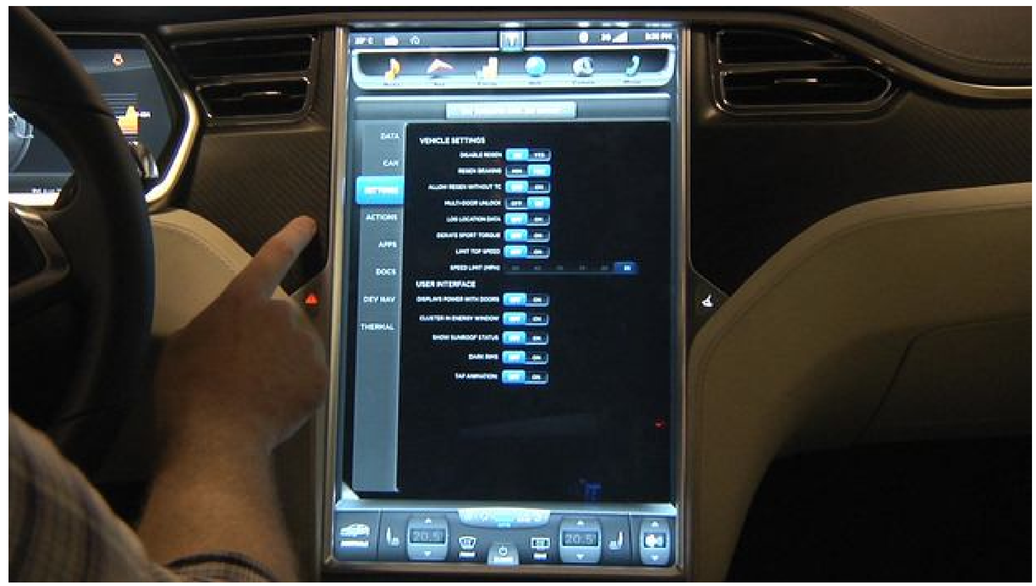
I'll promise …

An Undecidable Problem

The general problem of software verification **is not** solvable by computers.

Researchers show off remote attack against Tesla Model S

The researchers were able to remotely control the braking system, sunroof, door locks, trunk, side-view mirrors and more.



http://www.computerworld.com/article/3121908/security/researchers-show-off-remote-attack-against-tesla-model-s.html