Software Specifications
Uncomputable Problems For Strings

Cain Susko

Queen's University
School of Computing

April 4, 2022

Motivation

While we have explored the unsolvability of models like general programming languages (ie. C) we have not explored it in other models like for example, the simpler model of a Context free grammar.

Post Correspondence Problem (PCP)

Given the input:

$$(u_1, u_2, ..., u_n), (v_1, v_2, ..., v_n)$$

such that:

$$u_i, v_i \in \{a, b\}*, i = 1, ..., n$$

The **Question** is, does there exist $i_1, i_2, ..., i_k \in \{1, ..., n\}$ such that:

$$u_{i1}, u_{i2}, ..., u_{ik} = v_{i1}, v_{i2}, ..., v_{ik}$$

The indices $i_1, ..., i_k$ do not need to be distinct and there is no upper bound for k. The Turing machine halting problem reduces to PCP witch means PCP is uncomputable.

Context Free Languages

Using reduction from PCP, it can be shown that many decisions problems for CFG are unsolvable. Some of the following problems are unsolvable:

- Equivalence - do the two given CFGs generate the same language?
- does a CFG generate $\Sigma*$?
- is a given CFG ambiguous?

Note: CFGs are useful because they can be parsed efficiently!

Regular Languages

A lot of problems pertaining to CFGs are unsolvable, but perhaps regular languages, being more restrictive, may have less. In fact, all 'natural' problems for regular languages are solvable (in principle). These include equivalence, minimization, etc...
There exists counter examples to this that have a somewhat 'un-natural' definition. For example:

Example Choose X to be a fixed language (with a complicated definition), the **Question** is: for a given Deterministic Finite Automaton A, do we have:

$$L(A) \subseteq X$$

Note: with a suitable definition of X, the question is unsolvable.

Computational Complexity

Just because something is solvable, does not mean it is solvable within the time on the universe. We must be aware of the computational complexity of the algorithm which solves a problem. The existence of an algorithm does not mean the algorithm can actually be used.

Example Integer Factorization is trivially solvable using brute force, however, the larger the number is the longer the computation will be, which at a certain point, makes the problem unsolvable.

Regular Languages DFA equivalence and minimization have efficient algorithms (polynomial time). In contrast, NFA or regular expression minimization and equivalence problems are uncomputable (intractable).