Airport Security, Generalized Chess, and $\mathsf{NP} \neq \mathsf{P}$

Albert Gao / sixiangg@andrew.cmu.edu

Abstract

One of the timeworn traditions of coursework at Carnegie Mellon University is that of bonus homework or exam problems, so as to stimulate the motivation and diligence deeply hidden inside the student psyche. A relatively straightforward instance of such exercise given to first-year undergraduates is the problem of $\mathsf{NP} \stackrel{?}{=} \mathsf{P}$. Most intelligent and studious first-years are able to answer immediately in the negative and give "a truly marvelous demonstration of this proposition which this margin is too narrow to contain"[1]. Here we present an alternative method for the same problem. We stress that our approach, albeit less simple than existing techniques often discovered by brilliant students, can be interesting by itself and be applied to other problems.

1 Introduction

Originally proposed by [Coo71], the $\mathsf{NP} \stackrel{?}{=} \mathsf{P}$ problem often needs no introduction, since frequently it is given as an exercise in introductory computer science courses. For those familiar with the rich zoo of complexity classes, this section may be skipped.

We briefly give reminders of some definitions here so our later results are consistent with the notation. The class P is the class of languages L for which there exists a deterministic polynomial time Turing machine M such that $M(x)$ gives 1 if and only if $x \in L$. The class NP is the class of languages decided by polynomial time Turing machines in the nondeterministic model. Equivalently, $L \in \mathsf{NP}$ if and only if there exists a deterministic machine M with the following two conditions.

- For any $x \in L$, there exists poly-length y with $M(x, y) = 1$.
- For any $x \notin L$, for any poly-length y we have $M(x, y) = 0$.

The complexity class $\mathsf{EXPTIME}$ is the set of all languages decided by deterministic Turing machines in time $O(2^{p(n)})$, where $p(n)$ is a polynomial function in n. Note that this class of problems is crucial for our method, even though most usual solutions skip this definition and derive contradictions from $\mathsf{P} = \mathsf{NP}$ directly.

2 Preliminaries

Here we mention several well-known results.

Lemma 2.1. $\mathsf{AS} \in \mathsf{NP}$.

Proof. The problem of Airport Security (AS) is a classic example of a problem in NP. As in Figure 1, AS is a decisional problem that attempts to distinguish a person conducting terrorist activities from a person not conducting terrorist activities.

[1] Folklore indicates that the common approach to this exercise employs machinery from Fermat's solution to some archaic number theoretic proposition outlined in the margins of one of his books.

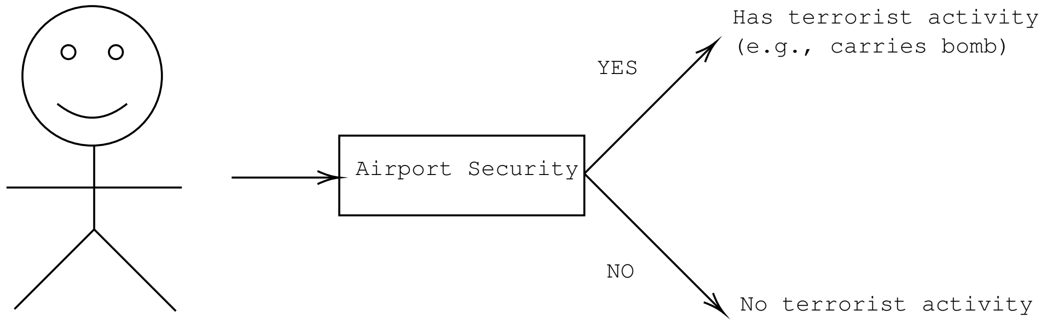

Figure 1: Illustration of problem AS.

The act of searching gives us a natural candidate for the witness string. That is, the terrorist act itself serves as y in our previous definition. If a person is a terrorist, we have a short proof of it that is, for example, a scanned image of a C-4 explosive. On the other hand, if a person is not a terrorist, it is difficult to succinctly prove this fact—we still need to search this person from toenails to hair strands anyway. But since there is no proof of terrorism for non-terrorists, this problem is naturally in NP. □

Lemma 2.2. GC *is* EXPTIME-*complete.*

Proof. Generalized Chess (GC) is shown to be EXPTIME-complete from [FL81]. □

3 Main Theorem

Theorem 3.1. $\mathsf{EXPTIME} = \mathsf{NP}$.

Proof. We establish that $\mathsf{EXPTIME} \leq \mathsf{GC} \leq \mathsf{AS}$, which shows the claim since $\mathsf{NP} \subseteq \mathsf{EXPTIME}$. Taken a GC instance and we are to determine if one party (call it "white") can force a win, we can have a diplomat Bob write out a proposal detailing a suggestion for a diplomatic game between two government parties according to the given GC instance. That is, we enforce the rules and positions specified in GC and have the enemy king be a political figure whose demise constitutes a terrorist act. A winning GC therefore corresponds with the diplomat's detailed plan for a terrorist act, yet a non-winning GC represents a friendly diplomatic game suggestion. Hence we can ask such diplomat to print out the GC instance and go through the AS black box to resolve GC. □

Corollary 3.1.1. $\mathsf{NP} \neq \mathsf{P}$.

Proof. $\mathsf{P} \subsetneq \mathsf{EXPTIME} = \mathsf{NP}$. □

References

[Coo71] Stephen A Cook. The complexity of theorem-proving procedures. In *Proceedings of the third annual ACM symposium on Theory of computing*, pages 151–158, 1971.

[FL81] Aviezri S Fraenkel and David Lichtenstein. Computing a perfect strategy for n× n chess requires time exponential in n. *Journal of Combinatorial Theory, Series A*, 31(2):199–214, 1981.