Maynooth University (National University of Ireland, Maynooth)

Department of Computer Science

T. Naughton

CS605 : NP-completeness intro 1

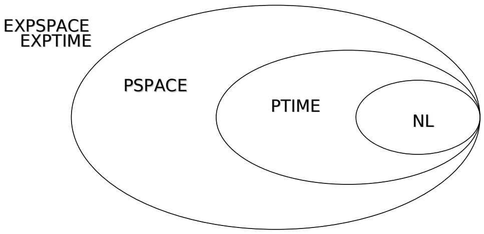
March 2020

NP-completeness

- Alternative introduction to the class NP and to NP-completeness

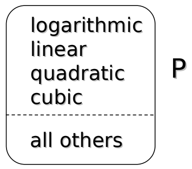
Where are the tractable problems?

- Where do tractable problems end and intractable ones begin?



Where are the tractable problems?

- Some problems in **P** are ‘practically’ intractable, so it would make sense that the dividing line between tractable and intractable problems should be somewhere in **P**.



A new class of problems

- However, to confuse things, it became evident by the late 1960s that some seemingly simple problems resisted polynomial time algorithmic solutions.

- Worse, it could <u>not</u> be proved that they were exponential either!

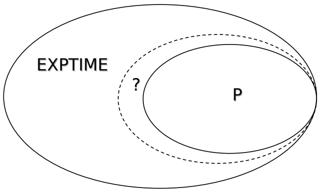
A new class of problems

- Was it simply because no clever person had found a polynomial algorithm yet?

- And in the meantime, where do these problems belong?

A new class of problems

- The net result was that the dividing line between tractable and intractable had to move *outside* **P**.



A new class of problems

- In an attempt to classify this strange new family of problems, Steven Cook (and Leonid Levin, independently) defined a new class of problems.
- Their goal was to distinguish this new family from *provably* intractable (i.e. provably exponential) problems.

Verifiable in polynomial time

- Cook made a very useful observation:

  given a problem in **P**, one should be able, at the very least, to **verify** a given correct solution in polynomial time.

Verifiable in polynomial time

- This makes sense, because when a correct algorithm solves a problem, then implicitly *it is providing a proof that the answer is correct*.

Verifiable in polynomial time

- It is likely to take only linear time to verify the correctness of a solution to a problem in **P**, but in the worst case all we have to do is rerun the whole polynomial algorithm to verify that the given solution is the correct one.
- If this <u>can't</u> be done for a particular problem in polynomial time, then the problem is definitely *intractable*.

Nondeterministic polynomial time

- Thus the family of problems for which one can verify the solution in polynomial time was born.
- The class was called *nondeterministic polynomial time*, or NP.

Nondeterministic polynomial time

- The reasoning was,

  in this class we are saying nothing about how the answer was found,

  it has been found in a **nondeterministic manner**,

  but we can verify that our answer is the correct one in **polynomial time**.

A common misconception

- NP does not mean *non-polynomial* :
- As we have discussed, all problems solved in polynomial time are verifiable in polynomial time, so P is a subset of NP.
- To call NP "non-polynomial" is obviously wrong, because it does contain polynomial-time problems.

Aside:

- The new class of problems (such as TSP) that was neither provably tractable nor provably intractable was given the name NP-complete (justifaction given later).
- These problems are in NP, and (provisionally) outside P.
- They can appear to be very different from each other.
- They are related to each other in a very interesting way.

Aside:

- Whether P is a *proper* subset of NP is an open question.
- If a single one of the NP-complete problems is *proved* to be either in P or not, then the question will be answered.
- This problem, the P =? NP conjecture, first raised in the early 1970s, remains one of the most difficult unresolved problems in computer science.