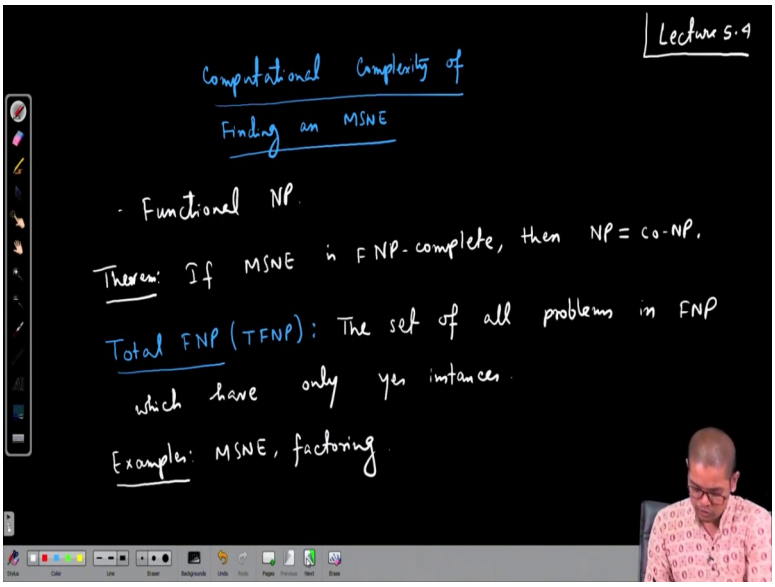
Algorithmic Game Theory
Prof. Palash Dey
Department of Computer Science and Engineering
Indian Institute of Technology, Kharagpur

Lecture - 24
PPAD Class

Welcome, so in the last lecture we started discussing the complexity of the problem of finding MSNE and we will continue the discussion in this lecture.

(Refer Slide Time: 00:39)

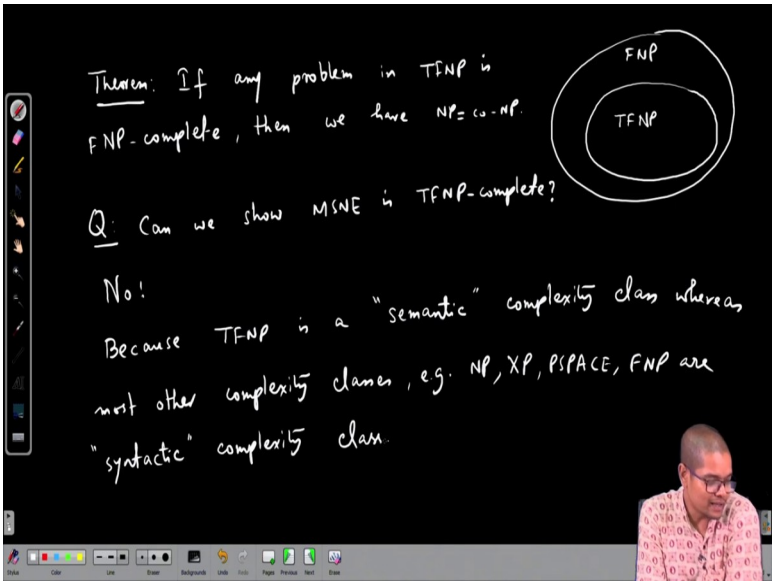


So, computational complexity of finding an MSNE and we discussed that why NP cannot capture this problem because it is not a decision problem and then we describe the complexity class called functional NP. Which is same as NP, but you know for yes instance we need to output a certificate. But we observed we proved this important theorem important, but easy theorem that if MSNE is FNP complete FNP complete then NP equal to co-NP which we consider very unlikely.

So, and why ok so intuitively speaking why MSNE is not FNP complete or may not be FNP complete, is that you know the problems in NP the heart of the difficulty lies in deciding whether it is yes or no. Which is not the case in in MSNE for MSNE the it is always is the heart of the difficulty is in finding a MSNE.

So, we define a functional subclass of another subclass of FNP which is called total FNP TFNP in short. So, what is total FNP? This is the set of all problems in FNP which have only yes instances only yes instances ok. So, examples are so what are the examples? Say MSNE problem factoring given a positive integer factor it into prime factors ok so ok.

(Refer Slide Time: 04:57)



Now, the next question so here is this this this how the picture looks like this is FNP functional NP inside FNP there is a subclass called TFNP. And you know this proof of the last theorem that you know if MSNE is FNP complete then it is we have NP equal to co-NP. If you carefully look at the proof it actually proves that this theorem. If any problem in TFNP is FNP complete, then we have NP equal to co-NP.

So, you do not expect any problem in TFNP to be FNP complete; that is why we do not expect to show factoring also to be FNP complete. But can we show this is to be TFNP complete. So, question can we show MSNE is TFNP complete the answer is again no, but you know why because you know TFNP we write because TFNP is a what is called semantic complexity class. Whereas, most other complexity class most other complexity classes for example, NP or say XP or PSPACE or FNP or syntactic complexity class.

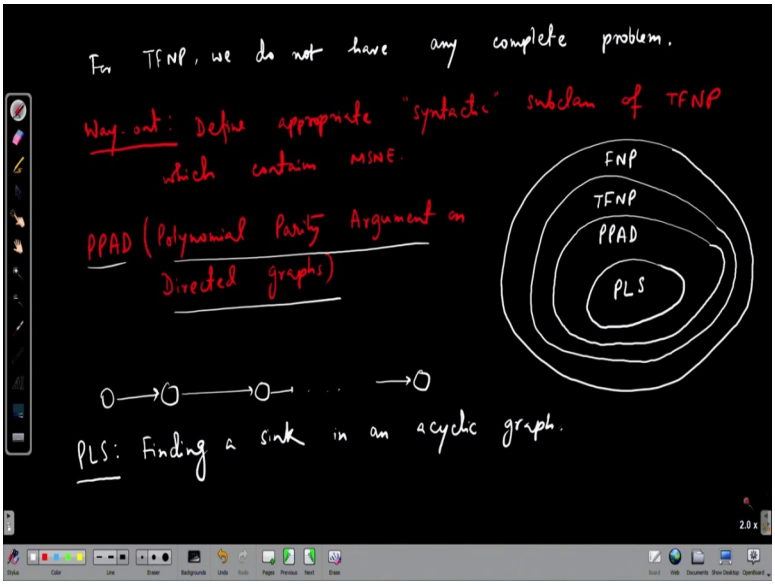
What we mean by syntactic versus semantic why. So, let us figure first understand what why what is syntactic complexity class what is NP? A problem is called NP a decision problem is called NP if it is accepted by non-deterministic turing machine. What is non-

determinist turing machine? It is a computational model. So, certain kind of machine should be able to perform certain kind of operation for a problem to belong to this syntactic complexity class like NP or XP or FNP like that.

On other hand semantic classes you know it is not defined with respect to some one machine or some abstract machine. For example, why MSNE belongs to TFNP? Why because of NASH theorem because of for some real analytical reason that every finite game has a mixed strategy NASH equilibrium that is why MSNE belongs to TFNP complexity plus.

Why factoring belongs to TFNP complexity class, because of some number theoretic reasons. So, if you pick any problem in TFNP they belong to TFNP because of some various reasons not one syntactic reason. So, various other reasons and that is the main idea why it is called a semantic complexity class and for semantic complexity class we do not know any complete problem, means forget proving that MSNE is TFNP complete.

(Refer Slide Time: 10:25)



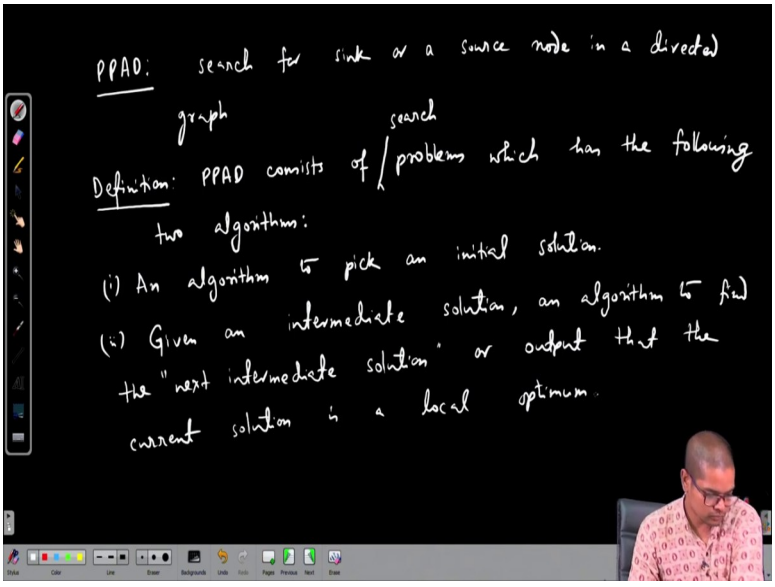
So, for TFNP we do not have any complete problem. So, we do not hope to show that MSNE is TFNP complete. Now, so what we do is that we so the way out define appropriate syntactic subclass of TFNP which contains MSNE so and that is what we do is PPAD. So, here is so what are the complexity classes let us draw we have PLS polynomial local search, we define FNP functional NP sorry not this those local sets are different.

So, here outside we have FNP functional NP then in within that we define TFNP and within that we have PPAD. So, that we will define and within that there is a complexity class PLS which we have already defined. So, let us define PPAD which is a semantic subclass of TFNP PPAD. What does this PPAD stand for? Polynomial Parity Argument on Directed graphs.

So, problems in PPAD, you know the problems in PPAD or let us see problems in PLS problems in PLS can be thought of traversing in the solution space or directed graph in the solution space and we are finding we are trying to find a sink node. So, PLS is abstractly finding a sink a node without degree 0 finding a sink PPAD is also finding a sink , but not but in a more general graph.

So, PLS is finding a sink in an acyclic graph. Now you say that you know finding a sink in acyclic graph is easy why problems in PLS is considered to be hard PLS complete, you know this graph is huge graph it is not given explicitly it is given by those 3 algorithms in PLS. So, that is why when you say polynomial it is not polynomial in the number of vertices in the graph.

(Refer Slide Time: 15:58)

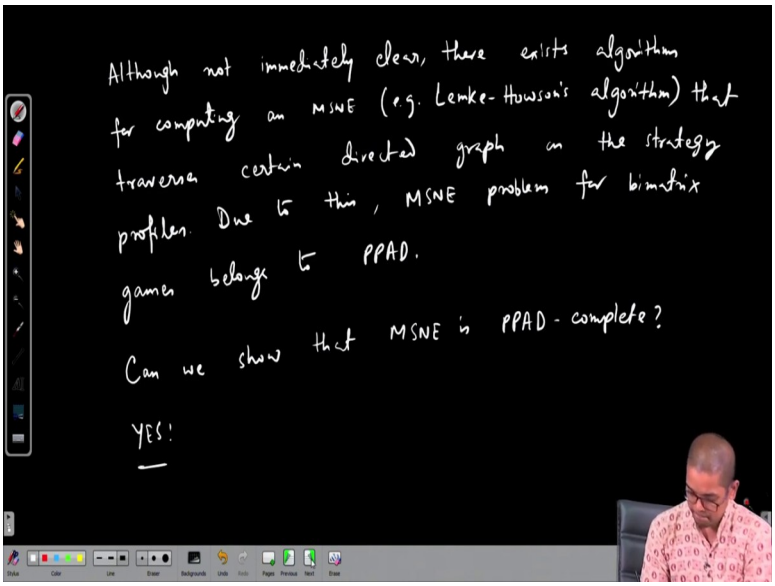


Similarly, PPAD is also searching in the searching for a sink node. So, let me write PPAD is or search for sink or a source node in a directed graph, again remember the graph is huge it is on the solutions on the strategy profiles for MSNE graph and the graph for PPAD can contain cycle it need not be a acyclic graph.

So, the class pp ad consists of problem which has 2 algorithms. So, now the formal definition let me write definition. PPAD consists of problems consists of search problems abstract search problems. What is abstract search problem? We have defined it is defined by 3 algorithms or an algorithm to find the initial solution and algorithm to find a value here the value is not needed and an algorithm to find the next solution.

So, PPAD consists of search a problem which has the following 2 algorithms. What is the first algorithm an algorithm to pick an initial solution to pick an initial solution and given an intermediate solution, an algorithm to find the next intermediate solution for output that the current solution is a local optimum ok.

(Refer Slide Time: 20:29)

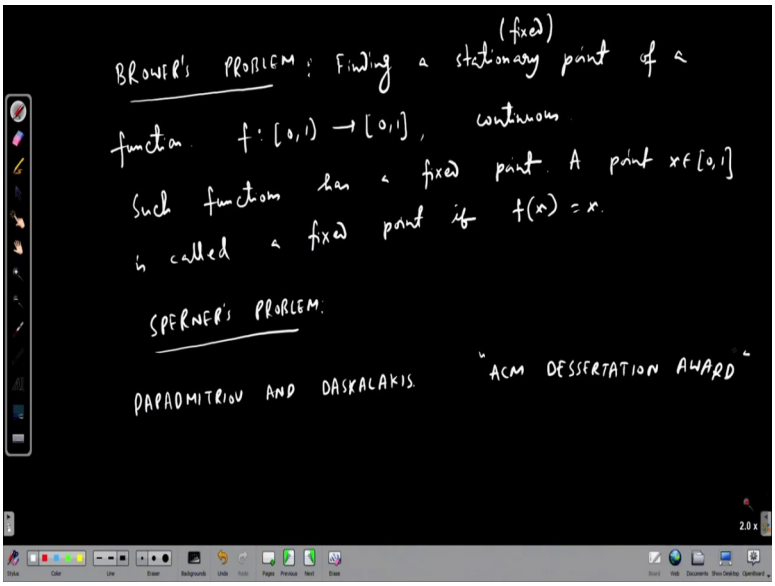


So, although it is not immediately clear. So, let me write. So, it is not immediately clear how MSNE can be viewed as an as a problem in PPAD, but indeed there exists an algorithm which traverses this director graph. Although not immediately clear there exists algorithms for computing an MSNE, for example Lemke Howson's algorithm that traverses certain directed graph on the strategy profiles.

So, due to this MSNE problem for bimatrix game belongs to PPAD ok. You see that you know this is a syntactic subclass of TFNP which is a semantic class, because you know again this the definition is abstract you know some there should exist some algorithms it is not that this MSNE problem belongs to PPAD because of some number theoretic reason or real analytic reason or something like that no it is not like that ok.

Can we show that this is PPAD complete, can we show that MSNE is PPAD complete? The answer is yes and the steps followed is closely resembles the original proof of Nash's theorem by which he showed existence of MSNE in any finite strategic form game.

(Refer Slide Time: 24:30)



So, with some intermediate we define some intermediate problems like Brower's problem which on a high level it the high level idea is finding a stationery point of a function ok. And then so there are functions like from say 0, to 0, 1 which is continuous and we know that this sort of functions a continuous function from 0, 1 to 0, 1 it has a fixed point stationary point or it is also called fixed point Brower's fixed point.

So, such functions has a fixed point a point x in 0, 1 is called a fixed point if this point is not moved by the function if f x equal to x. Then there is also some other intermediate problem which is called Sperner's problem which we will discuss in detail in the next class. But and all these things all these proofs are very non trivial and involved and very lengthy and these papers this is due to the due to Papadmitriov and Daskalakis this theory Papadmitriov and constant is Daskalakis and it was Daskalakis PHD thesis.

Papadmitriov was his PHD supervisor and it got the ACM Dissertation Award this thesis this work because of this work this thesis got the act ACM Dissertation Award. So, this just to motivate you know this is the high level idea and we do not go into the proof of this theorem this is beyond the scope of this course.

But we will see a beautiful Sperner's Lemma in the next class and we will also and that is it. So, we will see beautiful combinatorial proof and that also give a high level glimpse of how this this this proof goes and we will finish with finding a with seeing a concrete problem concrete algorithm for epsilon PSNE in the next class ok.

Thank you.