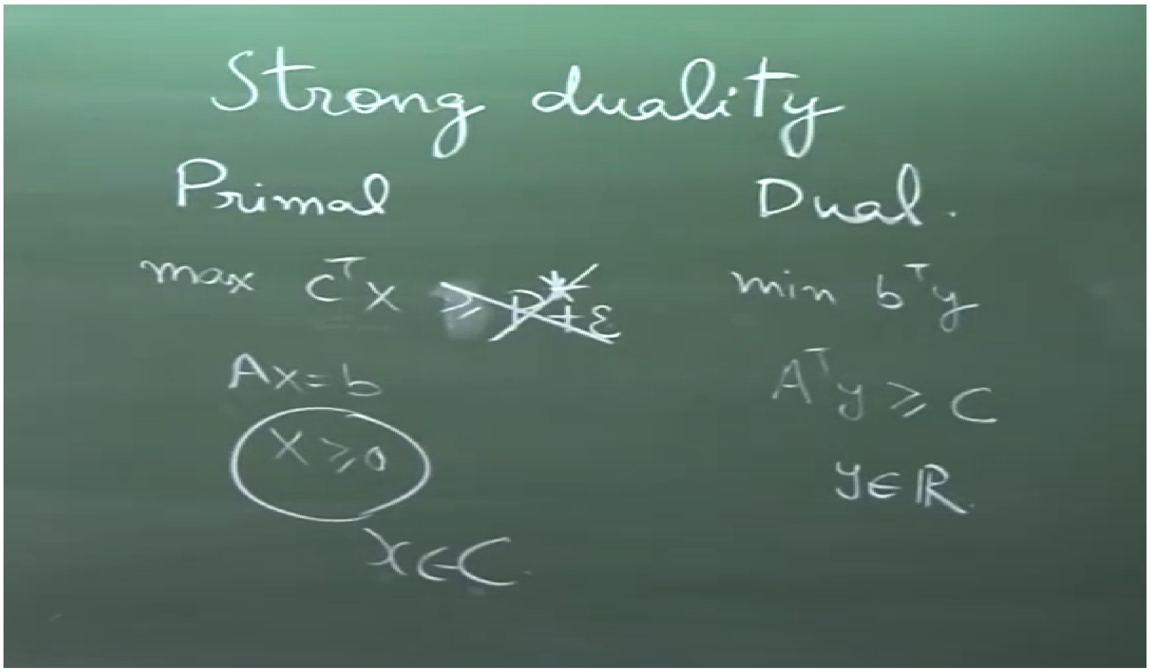
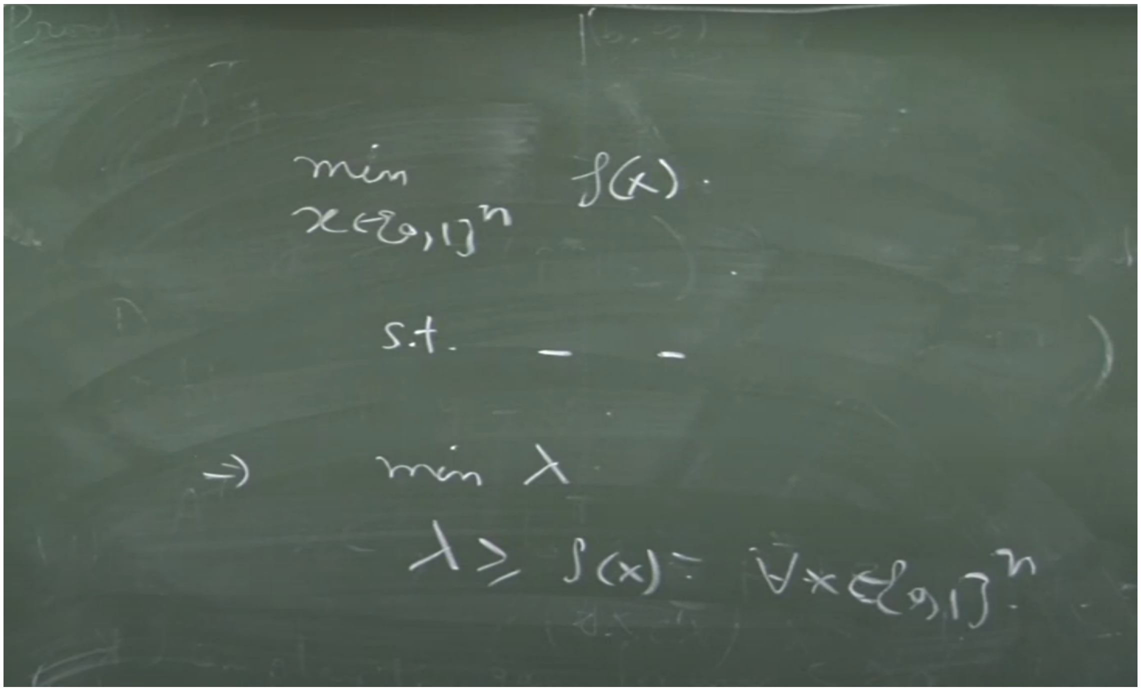
**Linear Programming and its Applications to Computer Science**
**Prof. Rajat Mittal**
**Department of Computer Science and Engineering**
**Indian Institute Of Technology, Kanpur**

**Lecture – 30**
**Complementary Slackness**




The idea is very simple. We look at this, we say that this is going to be hyperplane separation theorem just in one more dimension. When we think of this as one more constraint and then start applying, there are few technicalities which remember which kind of correspond to your geometry being weird. You take them out and you can just directly create a dual solution. And this has a nice interpretation, this is a nice ring to it. If you think about it, this is basically saying that your dual is a hyperplane separation in one more dimension.

And this trick of using the objective function and putting it as a constraint, you will see in many, many cases. In an application sometimes is helpful to kind of say, let us say you have some function of x and you have some constraints such that right now, this does not look like a single value which you are minimizing. Does not look like a linear program or does not look like the usual program.

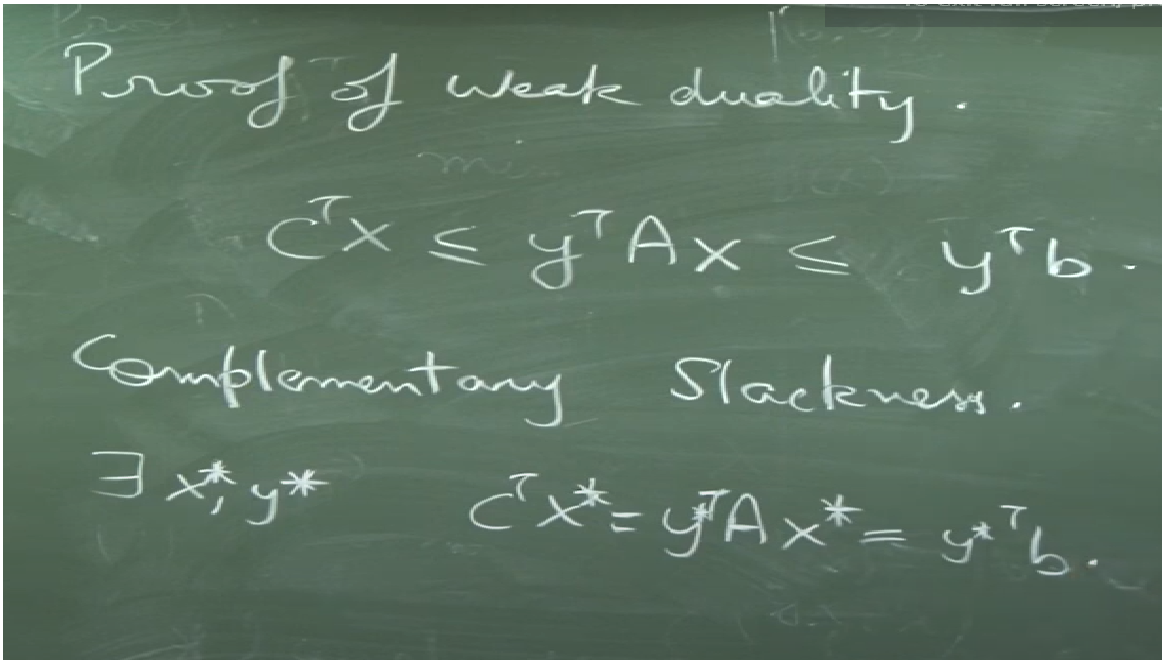



What you can do is, you can say minimize lambda such that lambda is greater than f x. And those kind of tricks lambda I think less than maximize lambda lambda less than f x one of them. But, these techniques are used to convert your objective function as a constraint. So, anytime someone is worried why strong duality happening, it is because of hyperplane separation in one more dimension. And I think hyperplane separation does not feel very weird to you.

In geometry at least you have a convex body, you have a point outside it, there has to be hyperplane separating it. That is the essential reason why strong duality works. Once more this was just for example, it could be any other standard format you will still get an analogous theorem. So, never get hung up on standard format of linear program. In any application you will never get a standard format.

The standard format is only helpful to prove things about LP. But, when you actually manipulate LP's there always inequalities, equalities, mix of them everything and we deal with them as it is. So, I think getting hung up on standard format is a bad idea, sounds good. So now, one of the kind of the nice thing about strong duality which I wanted to discuss was if this happens it actually gives us nice conditions to recognize whether your solutions are optimal or not. What do I mean by that? What was the proof of weak duality? The meta inequalities.

I think just to write in the same order. And now to change things actually what I want to talk about is something called complementary slackness. And it will be nice if these 2 programs are in the similar format. So, what I am going to do is change it to now this is also a standard format. A X less than equal to B X less than equal to 0.




What will change in the dual? What will change in the dual? A transpose? No. Fundamental principle if you change the constraint here, what will change here? The variables will change here. Remember, constraints correspond to variables here, variables correspond to constraints here. So, I have changed the constraint here that means what you have to change is this. So, what should come here? You can check A transpose sorry.

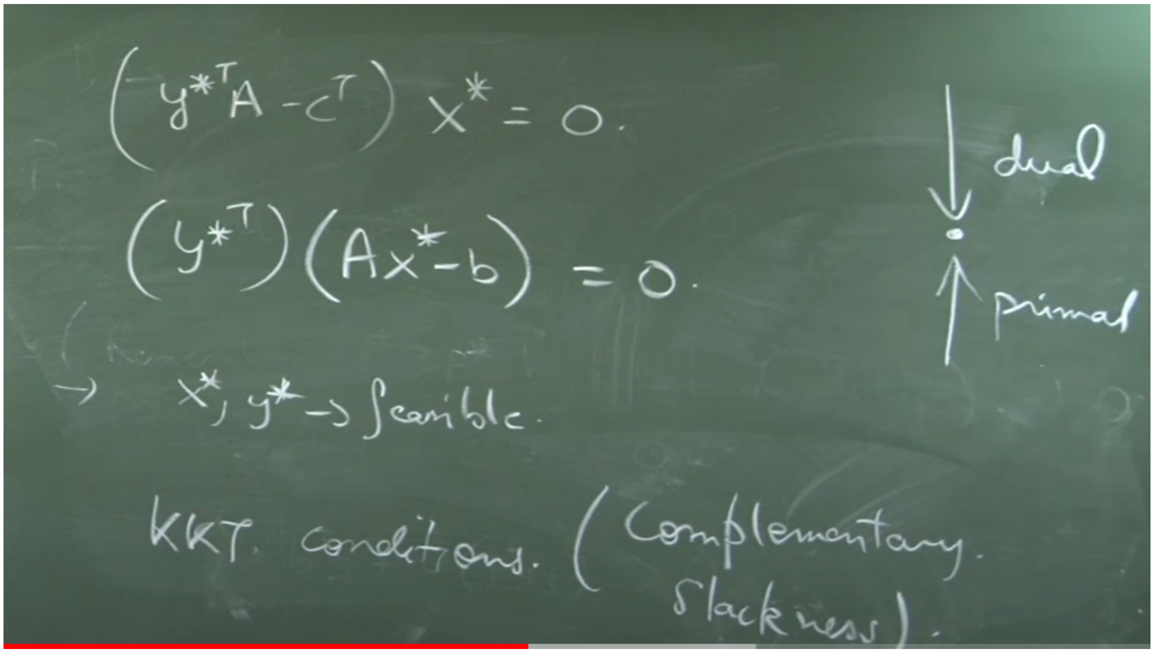
So, this is about this if A X is less than equal to B that means A is equal to B is equal to 0. So, this is my standard format. This is my proof of weak duality. But I know strong duality happens. That means assuming everything is feasible and all there exists X star Y star.

These are all equalities. So now, what do I know with this? I know I have just written this equality in a different format Y star A minus C times X star. You again see constraints of dual multiplied by variables of primal. Similarly, constraints of primal multiplied by variables. Now, what you know is that if X star and Y star are optimal solutions, then these things will happen.

What if X star and Y star satisfy these conditions? What do we know? Sorry, they need not be solutions. So, I have to add an extra thing. Suppose, I know that they are feasible. Then, I know that they are optimal solutions also. Confusion? Take it slow.

First clearly if X star and Y star were feasible and these two conditions were satisfied, they were the optimal solutions. Why? Because the values cannot improve. My dual values are going like this. My primal values are going like this. Any time they match, I am done.

I know that it is optimal. Any time I have constructed a dual solution and a primal solution such that their values match, I am done over. Those are optimal solutions. So, this gives me a way to figure out optimal solutions. This is a verification that your solution is optimal.

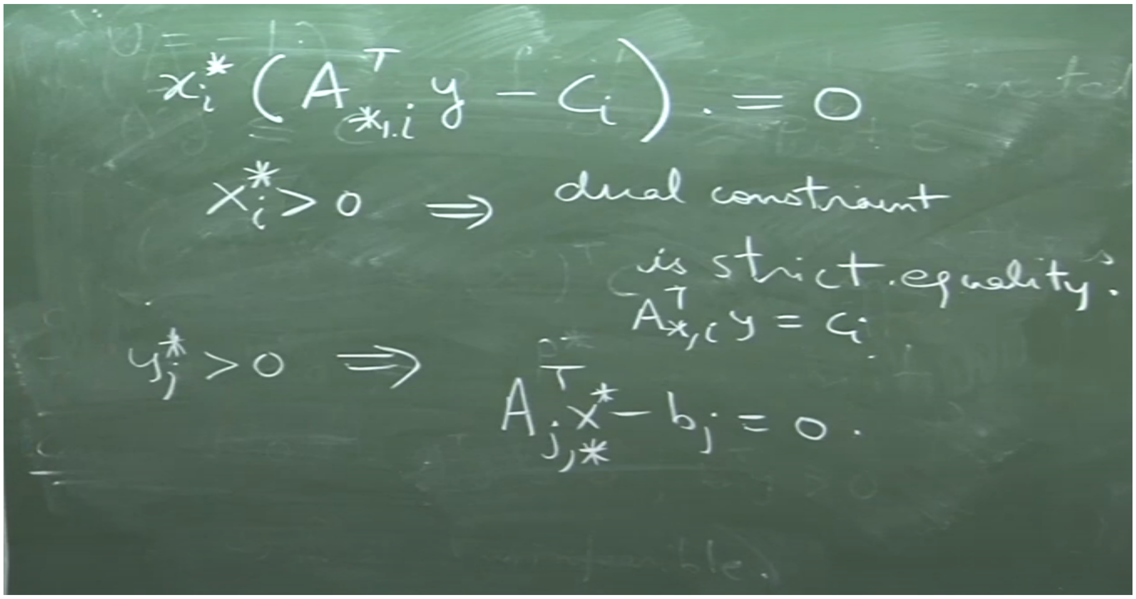



These are called KKT conditions in general for convex programming. For general convex programming, you can come up with KKT conditions. For linear programming, they are generally known as complementary. If X star and Y star are optimal, then this has to be satisfied with equality. That means, this has to be satisfied equality.

That means, KKT conditions will be satisfied. More importantly, if KKT conditions are satisfied, then these are actually optimal. This is the good thing about KKT conditions. We will be using them to solve a linear programming. I will give you an approach of solving linear programming by using KKT conditions.

These are three scientists Khan, Krush, Tucker. On these names, these conditions are different. Good. But, why is it called complementary slackness? What does it tell me? If my optimal primal variable is non-zero, suppose what is this? This is a vector equation. This is saying, if my X star is positive, then the corresponding variable has to be 0.

See all these, if I look at any particular column. This is my i th variable and this is my i th constraint in the dual. Notice, everything has to be positive in this. That means, all the terms in the summation are 0. That means, if X i is positive implies or similarly, strict means equality.




Generally, whenever say inequality is strict, I see. Dual constraint is strict means, I see the confusion. Similarly, if Y j star, then the corresponding primal constraint, which will be the j th constraint will be equal to 0. Good question, probably no.

No, it is a vector. That is why transpose is. I have to sit down and make sure, which one is transpose, which one is not transpose. But, I am sure all of you now are grown in up to figure out dimensions and whether there is a transpose or not. I do not want this course to be all about whether there is a transpose or not. It is basically a sanity check nothing else.

Questions? This is why it is called a nice complementary slackness condition. If my primal variables are non zero, it tells me about dual constraints. In some sense, the idea and this is the idea of the primal dual algorithms that you start with the solution and you see this is positive. Then, you can solve another dual where all these constraints are removed.

They are equality. You can simplify your dual. Using primal solution you try to create a dual solution. Using that dual solution, you will improve upon your primal solution and this feedback loop. Again, this is very meta approach.

You will see the exact approach. But, these are the ideas. Again, yes strong duality only says that these two values are equal. But, that is not just it. It is a very strong statement. It is giving me relationship between primal constraints and dual variables.

It is giving me complementary slackness. It is giving me lot of things in the background which now you should know. You should never think of strong duality just as saying these two values are equal. There are lot of things going on in the background.