Probability & Computing

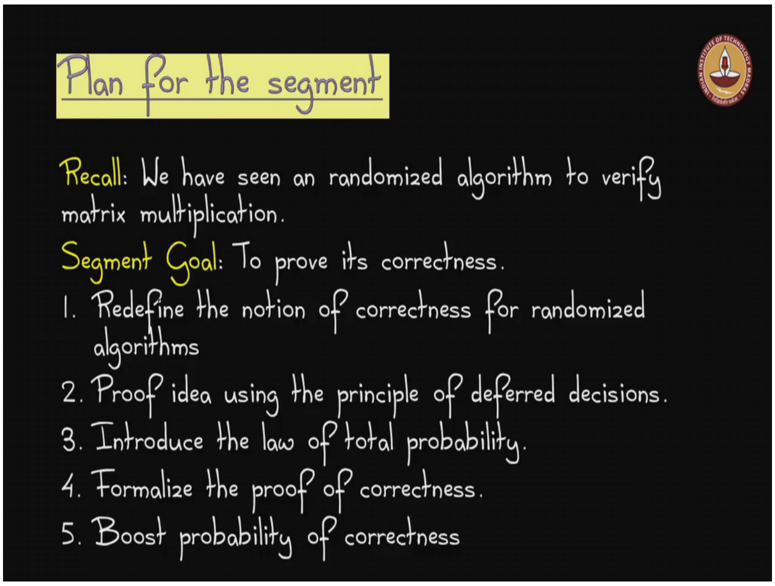
Prof. Jhon Augustine
Department of Science and Engineering
Indian Institute of Science, Madras

Module - 01
Introduction to Probability
Lecture – 04
Segment 4: Verifying Matrix Multiplication (Corrections & Law of Total Probability)

Let us get started with Module 1, Segment 4. So, we are going to continue our discussion of Verifying Matrix Multiplication. We saw an algorithm in the previous segment and today what we are going to do is prove that that algorithm is correct and the some definition of correct. So, we will have to redefine the notion of correctness for randomized algorithms system little bit at least some types of randomized algorithms.

And so, along the way we would be introducing a few things principle of differed decisions the law of total probability. So, these 2 principles we will discuss them as we go along.

(Refer Slide Time: 00:55)



Plan for the segment

Recall: We have seen an randomized algorithm to verify matrix multiplication.

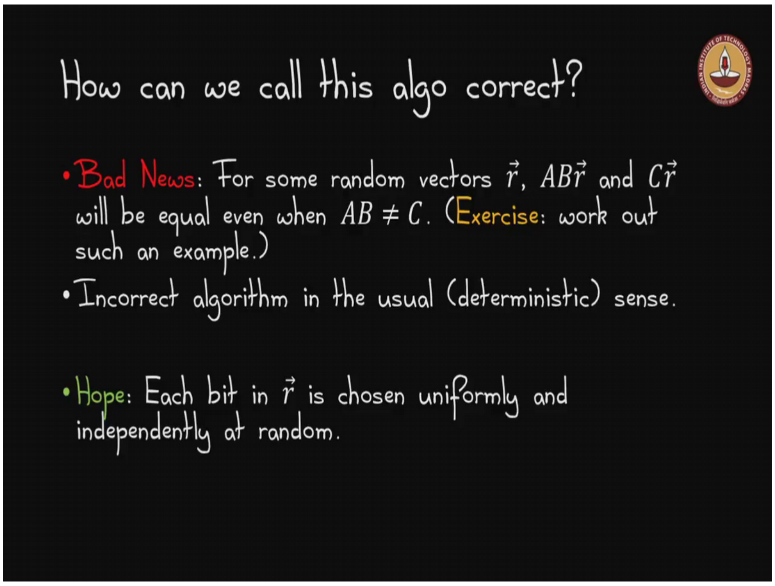
Segment Goal: To prove its correctness.

1. Redefine the notion of correctness for randomized algorithms
2. Proof idea using the principle of deferred decisions.
3. Introduce the law of total probability.
4. Formalize the proof of correctness.
5. Boost probability of correctness

And then will give A form of proof of correctness and initial proof of correctness will be a probabilistic statement.

So, it will come with certain probability guarantees, what will do is then will show how to boost the probability of correctness to pretty much any extent that we want ok. So, that is the plan for today's segment.

(Refer Slide Time: 01:16)



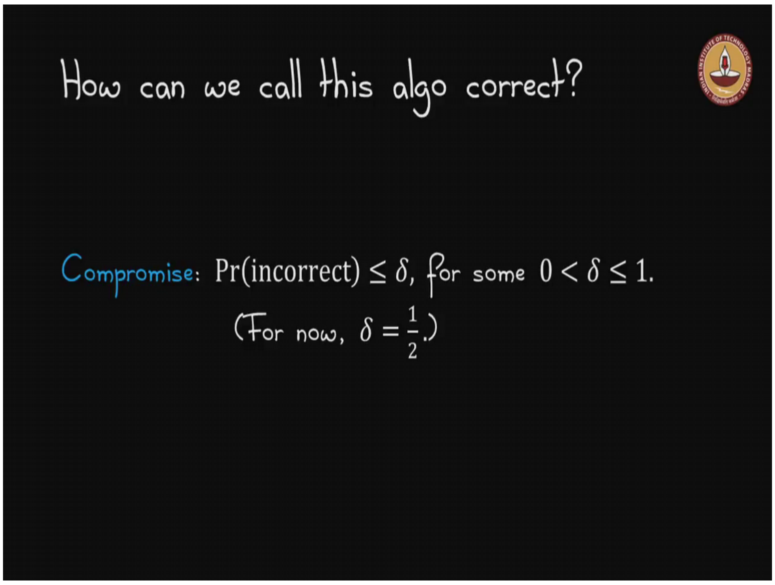
How can we call this algo correct?

- Bad News: For some random vectors $\vec{r}$, $AB\vec{r}$ and $C\vec{r}$ will be equal even when $AB \neq C$. (Exercise: work out such an example.)
- Incorrect algorithm in the usual (deterministic) sense.
- Hope: Each bit in $\vec{r}$ is chosen uniformly and independently at random.

Let us start with some bad news first hopefully to you know set ourselves the stage to give some good news later. So, if you recall the algorithm you chose a random vector r you multiplied A B r on the left hand side and then right hand side we might be multiplied C r, and we checked if they were equal they were equally reported A B was equal to C otherwise not. So, that is the algorithm.

And if you deterministically choose r you can always find an r for which the algorithm is going to be incorrect. So, an undergraduate algorithms course would not suffice to handle this ha problem, because then you will immediately think about the worst case and in the worst case this algorithm is incorrect at least in the deterministic worst case sense, but then our hope lies in the fact that r is chosen uniformly at random and each bit is independent of each other.

(Refer Slide Time: 02:09)



How can we call this algo correct?

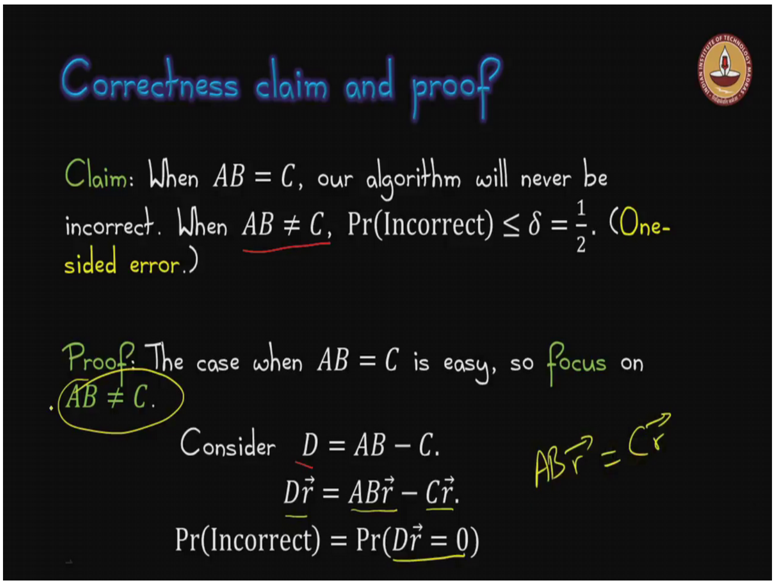
Compromise: Pr(incorrect) $\leq \delta$, for some $0 < \delta \leq 1$.

(For now, $\delta = \frac{1}{2}$.)

And so, we are going to redefine the notion of correctness and what we are going to do is prove the following state that remember, when you run the algorithm there are 2 outcomes that in this case our sample space is correct versus incorrect.

And we want to argue that the probably that the algorithm is incorrect is at most some delta in some delta and in this case we will start with delta being just a half. So, to begin with half the only guarantee we give is that half the time our algorithm will be correct, but the other half possibly could be wrong and then we will see how what we can do later with this ok.

(Refer Slide Time: 02:54)



Correctness claim and proof

Claim: When $AB = C$, our algorithm will never be incorrect. When $AB \neq C$, $\Pr(\text{Incorrect}) \le \delta = \frac{1}{2}$. (One-sided error.)

Proof: The case when $AB = C$ is easy, so focus on $AB \neq C$.

Consider $D = AB - C$.

$$D\vec{r} = AB\vec{r} - C\vec{r}.$$

$$\Pr(\text{Incorrect}) = \Pr(D\vec{r} = 0)$$

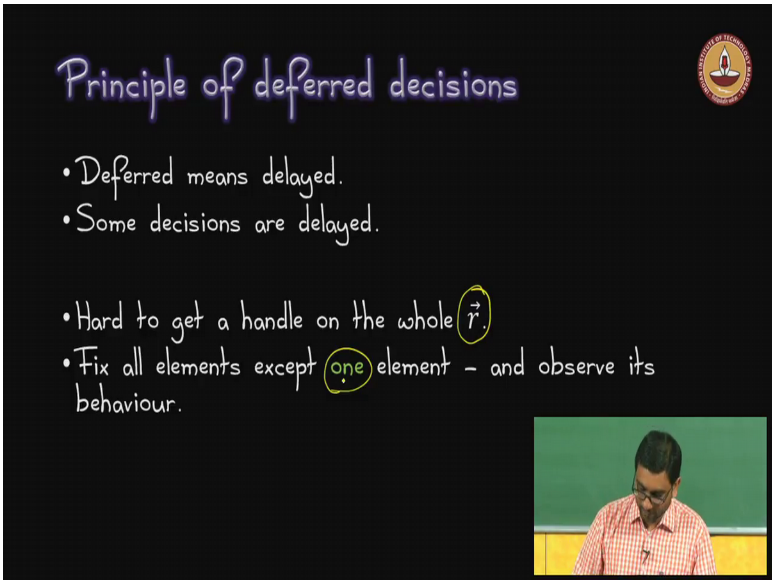
$AB\vec{r} = C\vec{r}$

So, since let us look at it a little bit carefully when A B is equal to C our algorithm will never be incorrect, why is that because whatever r you choose A B times r will equal C times r, because A B and C are equal. So, we are always going to be correct when A B equals to C the only interesting part therefore, is when A B is not equal to C and our algorithm if it was correct should be able to say that A B is not equal to C.

What we left hand show is that given that A B is not equal to C our algorithm will say incorrect with probability at most delta, that is what we are trying to for the rest of this we will be therefore, focusing on A B not equal to C we will just assume that A B is not equal to C ok. Now we define D equals A B minus C and therefore, now we can think of D times r equals A B times r minus C times r.

So, remember our algorithm is going to check if A B r is equal to C r. So, the other an equivalent check is to see whether D r is equal to 0. And if it was if it shows up to be 0 it is incorrect. Why because we have made the assumption that A B is not equal to C and D r equal to 0 would be fooling us into thinking A B is equal to C.

(Refer Slide Time: 04:26)



Principle of deferred decisions

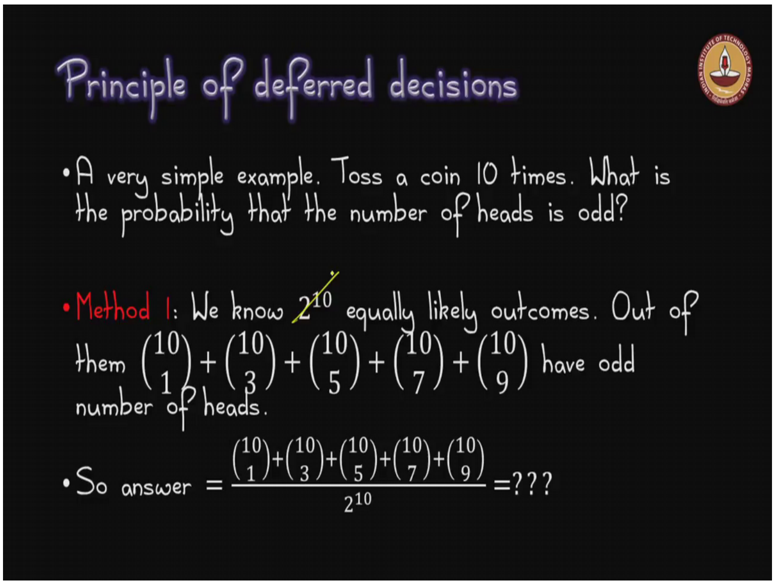
- Deferred means delayed.
- Some decisions are delayed.

- Hard to get a handle on the whole $\vec{r}$.
- Fix all elements except one element – and observe its behaviour.

So, now we want to apply a principle called the principle of deferred decision it is just a technique and in case you did not know deferred means delayed ok. Some decisions are going to be delayed and others are going to assumed to have happened ok.

So, in this case why are we applying this principle of deferred decision we have to contend with r which is an n bit vector. And what we are the way we are going to do that is we are going to fix all elements except one. This one element were going to choose and defer the decision on that element alone among this n elements. And then we will argue those that will be the deferred element and then we will argue based on that deferred element that the probability is that of in being incorrect is at most half ok.

(Refer Slide Time: 05:20)



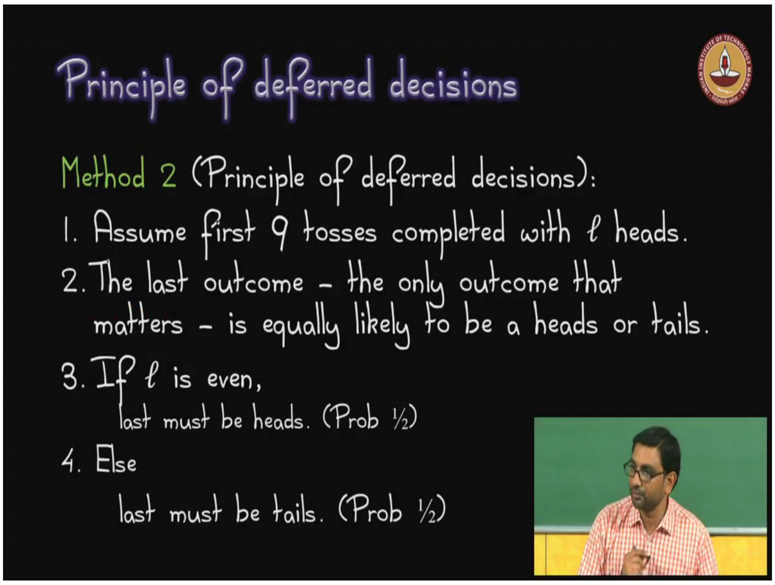
Principle of deferred decisions

- A very simple example. Toss a coin 10 times. What is the probability that the number of heads is odd?
- Method 1: We know 2^{10} equally likely outcomes. Out of them $\binom{10}{1}+\binom{10}{3}+\binom{10}{5}+\binom{10}{7}+\binom{10}{9}$ have odd number of heads.
- So answer $= \frac{\binom{10}{1}+\binom{10}{3}+\binom{10}{5}+\binom{10}{7}+\binom{10}{9}}{2^{10}} = ???$

So, let us illustrate the principle of deferred decision first by A simpler setting. So, let us reorient ourselves to the simpler example, toss a coin 10 times what is the probability that the number of heads is odd? This is the question at hand. So, here is one way we can do it we know that there are 2 power 10 equally likely outcomes, because there are n 10 coin tosses out of them, how can we how many outcomes have an odd number of heads? It is going to be 10 choose 1, plus 1 choose 3, plus 10 choose 5 and so on right.

So, the answer if you want to directly argue the answer for the number of heads being odd is you know you have all these 10 choose odd numbers and the summation of them in the numerator, and 2 raised to the 10 in the denominator and it is going to get a little messy to argue what this is were not going to bother about that.

(Refer Slide Time: 06:27)



Principle of deferred decisions

Method 2 (Principle of deferred decisions):

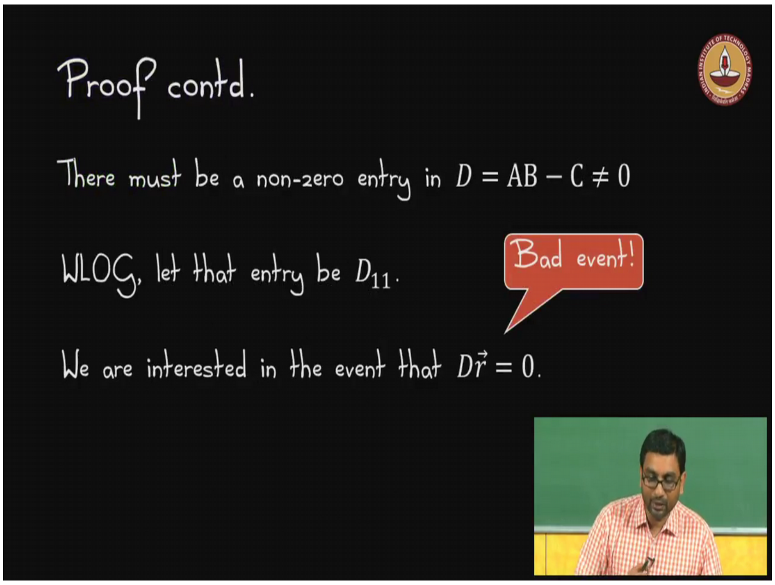
1. Assume first 9 tosses completed with ℓ heads.
2. The last outcome – the only outcome that matters – is equally likely to be a heads or tails.
3. If ℓ is even,
   last must be heads. (Prob ½)
4. Else
   last must be tails. (Prob ½)

But let us see how the principle of deferred decisions works in this case ok. Now what we're going to do is assume that the first 9 tosses have been completed and there are some l heads we do not know what that l value is there are some l heads ok?

So, now we are going to defer the decision for the last coin toss alone ok. And if you req if you think about it the last one is equally likely to be head or tails ok. And now you can see what happens if l is even ok, then how do we get an odd number of heads then the last will be heads with probability half, even if on the other hand is l is odd again last coin toss will be at tails with probability half.

And in either case we are going to be able to conclude at least intuitively at this point that the total number of heads is going to be odd with probability half ok. And better I mean this is just an intuition right now we will need the lot order probability to formalize this ok.

(Refer Slide Time: 07:35)



Proof contd.

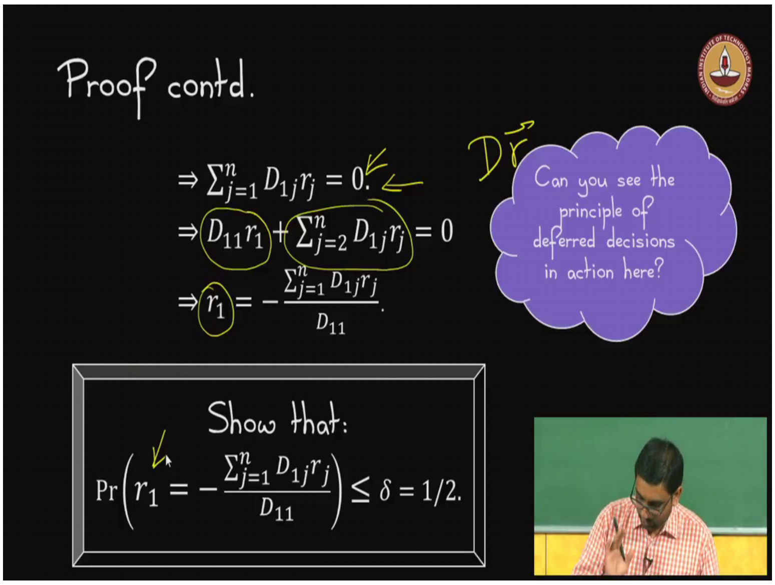
There must be a non-zero entry in $D = AB - C \neq 0$

WLOG, let that entry be D_{11}.

We are interested in the event that $D\vec{r} = 0$.

So, let us get back to the problem of matrix verifying matrix multiplication ok. So, now, let us look at D and we claim because remember A B D equals A B minus C and we are going to assume that A B is not equal to C. So, there must be A nonzero entry and were just going to assume that that nonzero entries is a very first one D 1 ok. And we are interested our remember the bad event, that we are trying to show the probability of that bad event is small is this D r equal to 0.

(Refer Slide Time: 08:17)



Proof contd.

$$\Rightarrow \sum_{j=1}^{n} D_{1j} r_j = 0.$$

$$\Rightarrow D_{11} r_1 + \sum_{j=2}^{n} D_{1j} r_j = 0$$

$$\Rightarrow r_1 = -\frac{\sum_{j=1}^{n} D_{1j} r_j}{D_{11}}.$$

$D\vec{r}$

Can you see the principle of deferred decisions in action here?

Show that:

$$\Pr\left(r_1 = -\frac{\sum_{j=1}^{n} D_{1j} r_j}{D_{11}} \right) \leq \delta = 1/2.$$

So, when we say D r equals to 0 we can just apply the formula and so, what we are going to do is work with the first row of D alone.

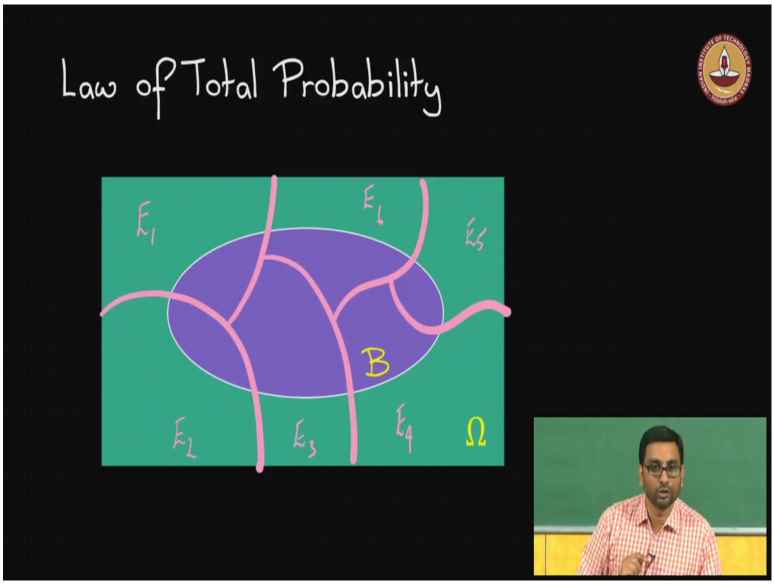
So, this is basically the first row of D multiplied with the elements of r and we want to ask how what is the probability that entry will be a 0 and that. So, it is if you think about it this is going to be one way in which the bad event one important requirement and a necessary condition for D r the matrix D times r to be equal to 0, basically this 0 is the first entry of D r ok. And we basically want to limit ourselves to proving that this itself is will happen with some bounded probability. So, now, let us expand this summation out.

So, it is basically what we are doing is isolating the first term and remember we want to use the principle of deferred decision. So, we were isolating the first term which is the one of the term that we are going to differ and the rest of the terms are here in the summation ok. And then we are going to were isolating just this r 1. So, we get some formula. So, for now focus on the fact that we have isolated r 1, this is the first bit in our random interval.

So, now, our statement can be re simplified as to show that the probability that the first bit r 1 equals this particular right hand side quantity is at most a delta ok.

So, it is slightly it seems messy, but what we have there the nice thing we have done is we have isolated our focus on just r 1. So, now, we have the ability to apply the principle of deferred decision. So, we do not we kind of try to avoid the other random bits we focus our efforts on see what happens to r 1. And so, that is the principles of deferred decision were going to try to apply.

(Refer Slide Time: 10:40)

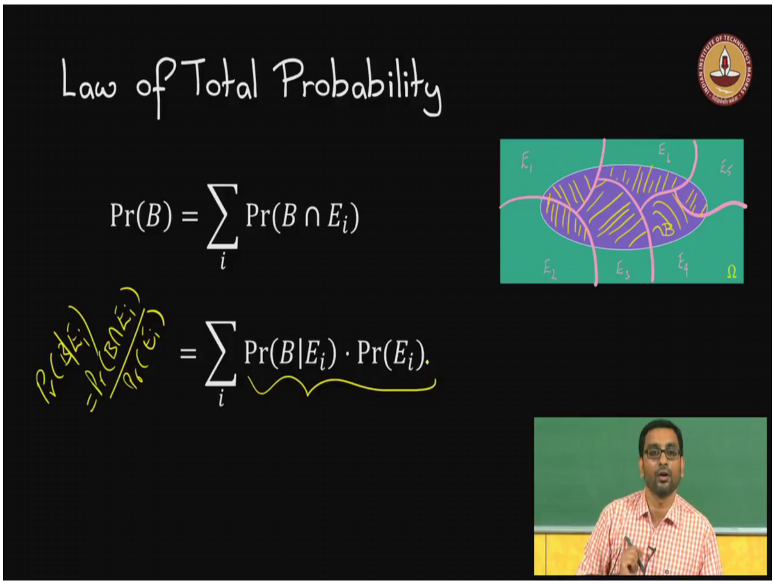


Law of Total Probability

But if you want to be careful and formal about it what we are actually doing is going to apply now at this point the law of total probability which will I will talk about now, how does this law of total probability work? Now assume that there is and this omega is a sample space and within that the sample space is broken into E 1 E 2 E 3 E 4 and so on. These are mutually disjoint subsets of omega and when you take the union of all these E 1 E 2 E 3 and so on, it should actually equal omega.

So, basically it is a partition of the sample space these E i's and our interest is a particular event B ok, that is the oval shown over here. And what the law of total probability says that now you can compute the probability of B by looking at the intersections of B with the E is ok.

(Refer Slide Time: 11:32)



Law of Total Probability

$$\Pr(B) = \sum_i \Pr(B \cap E_i)$$

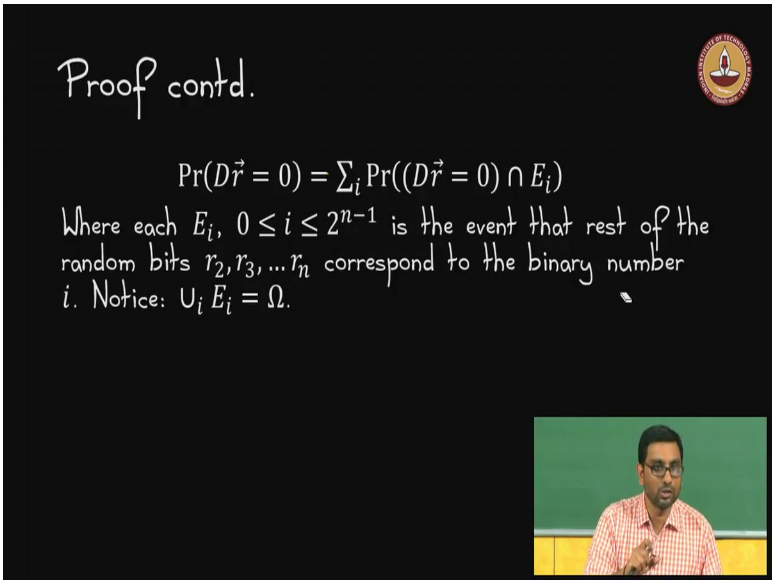
$$P(B|E_i) = \frac{\Pr(B \cap E_i)}{\Pr(E_i)} \qquad = \sum_i \Pr(B|E_i) \cdot \Pr(E_i).$$

And that is a if you look at this picture, it is a very intuitive thing to see right it is just when you intersect B with E 1 you get that portion of the sample space cover right.

And you just remember the axioms of probability theory you just have to add up these individual intersections and you get the probability of the event B ok. And now if you recall the formula for conditional probability what was that if you recall that it is going to be probability of B given E i is equal to probability of B intersected with E i divided by probability of E i.

And now we just take we a jiggle the terms around to get it to be in this form this is essentially the law of total probability, how are we going to apply it over here?

(Refer Slide Time: 12:47)



Proof contd.

$$\Pr(D\vec{r} = 0) = \sum_i \Pr((D\vec{r} = 0) \cap E_i)$$

Where each E_i, $0 \leq i \leq 2^{n-1}$ is the event that rest of the random bits $r_2, r_3, \ldots r_n$ correspond to the binary number i. Notice: $\bigcup_i E_i = \Omega$.

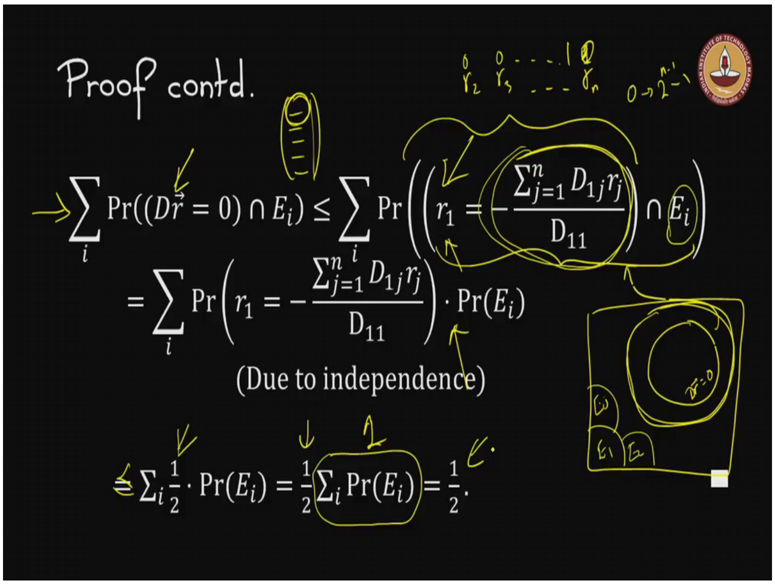
And remember we are interested in the our bad event is D r equal to 0. And so, that what is that probability, now you can take this bad event D r equal to 0 and you can intersect it with a bunch of E i ok. And if you can compute these individual intersection probabilities you can just add them up.

So, that is the law of total probability and what are the E is that we are going to take this is where it connects to the principle of deferred decisions. The E is are going to be the mean basically how many E i's are there are 2 raised to the n minus 1 E i's. So, I raise from 0 to 2 to the n minus 1 and it is even that the rest of the random bits correspond to the binary number i.

So, the first n n minus 1 bits or not the first 1 n minus 1 in this case the bits r 2 to r n can take on some binary value right. Remember and we are our intention is to not worry about any of them ok. And if you think about it the union of all of those events is going to be the sample space and they are mutually disjoint because you are talking about different random bits when, they when you evaluate them they value 2 different binary numbers ok.

So, now we can apply the principle of defer decision. So, now, what we are going to do is apply the principle deferred decision in this manner.

(Refer Slide Time: 14:20)



Proof contd.

$$\sum_i \Pr((D\vec{r}=0)\cap E_i) \le \sum_i \Pr\left(\left(r_1 = -\frac{\sum_{j=1}^{n} D_{1j} r_j}{D_{11}}\right)\cap E_i\right)$$

$$= \sum_i \Pr\left(r_1 = -\frac{\sum_{j=1}^{n} D_{1j} r_j}{D_{11}}\right)\cdot \Pr(E_i)$$

(Due to independence)

$$\le \sum_i \frac{1}{2}\cdot \Pr(E_i) = \frac{1}{2}\sum_i \Pr(E_i) = \frac{1}{2}.$$

So, we are basically now at this point isolating our focus to r 1, which we already showed we can do that and the thing remember the random bits were chosen independently. So, the whether r 1 equals to this quantity is going to be independent of the E is, because the e is depend on the rest of the random bits. So, we can simply because of that independence we can convert this intersection into a multiplication and we will get it in this form.

The first inequality let me be a little bit careful here this D r is the original bad event ok. That is the bad event that if you look at D r it is going to be a vector and it is going every element has to be a 0. What we are going to do is focus only on the first element and argue that just the probability of the first and that is what is happening over here, just the probability of the first element D 0 itself we are going to bound it. So, that is going to. So, this is going to be a stricter requirement this D r equal to 0, for simplicity we are going to bound so, if you think about the sample space.

So, let us draw the sample space D r the bad even D r equal to 0 is going to be something like this. So, this is a bad event D r equal to 0 ok. This will require all of them to be 0 what were going to do is instead focus on a larger event which only requires the first element to be 0 ok.

And we are going to argue the this is the this is this event and we are going to argue that this outer event itself is going to have small probability that is what we are doing that makes sense. So, if you look at each let us look at E 1 E 1 is going to be the event that if you take the random bits r 2 r 3 ok.

So, if you look at r 2, r 3 and so on up to r n they can take value 0 or 1. And now they can take how many values they can go to go from 0 to 2 to the n minus 1 and so on. These are all the possible ways that they can change and E 1 is the case where they take the value.

If you look at the by the rating they should, if you take this binary string where everything is here except the least significant bit that evaluates to a 1 right, that is E 1. It E 0 will be when all the bits are 0 E 2 will be when all the bits I mean the it will be something like 1 0 and so on. And that basically covers will cover the entire sample space and that is exactly what we wanted for the law of total probability.

So, that will in cover the entire sample space and were taking for each of the E i's were intersecting, it with this year this even that we care about this outer you know that is in the law of total probability figure that that is to be the oval be that we do. So, now, what we are going to do is just apply the formula the question is this how is this r 1 equal to this quantity, well r 1 is a bit value either it is either 0 or 1, and clearly let me make this why are these 2 events independent that is the question.

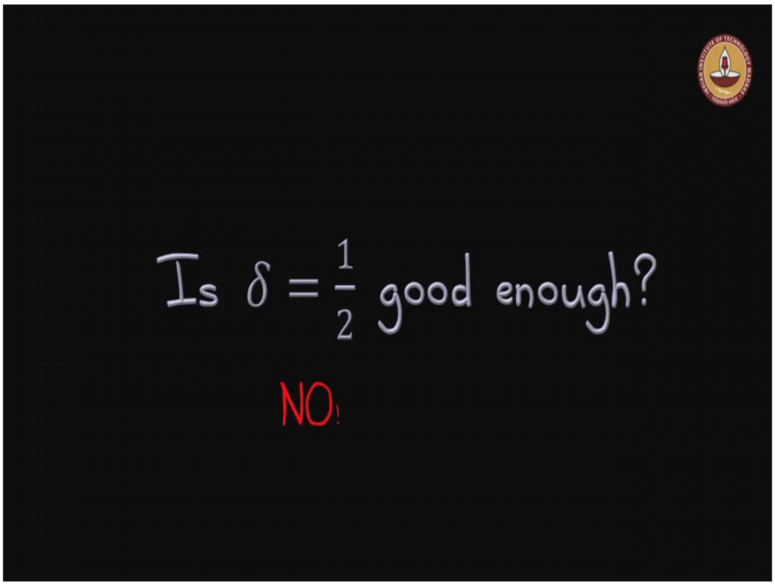
So, if you look at the first event r 1 what we have assumed is that we are working with the principle of deferred decision this right hand side has some quantity this right hand side has some quantity. And whether r 1 is going to equal that quantity or not is going to be completely independent of what other random bit values are and that is why r 1 this event is going to operate independent of E i's. The outcome of this event is going to depend purely on r 1 in the principle deferred decision we have at this point in time we have fixed this quantity that is the reason.

So, now what we are going to do is just apply some values. So, this r 1 how will it equal this quantity on the right hand side, well there are 2 possibilities r 1 is either 0 or 1. Certainly if it applies if r 1 is equal to this quantity on the right hand side when r 1 is 0, then when it is equal to 1 it would not be equal.

So, with probability at most half it is going to be equal that is where this half comes from and the probabilities of the E i's stay as us. So, we are take the half outside and well what is this probability this is basically events that span the entire sample space. So, that is equal to 1 which leaves us with the probability of half.

So, this is where how we get the fact that this the probability of this bad event is at most n r 1 is the index number 1 right, which is what we have taken it was an arbitrary choice, but we just fix assume that there was some it goes back to the fact that some entry in D was nonzero and we used that fact to work with one entry in r.

(Refer Slide Time: 20:28)



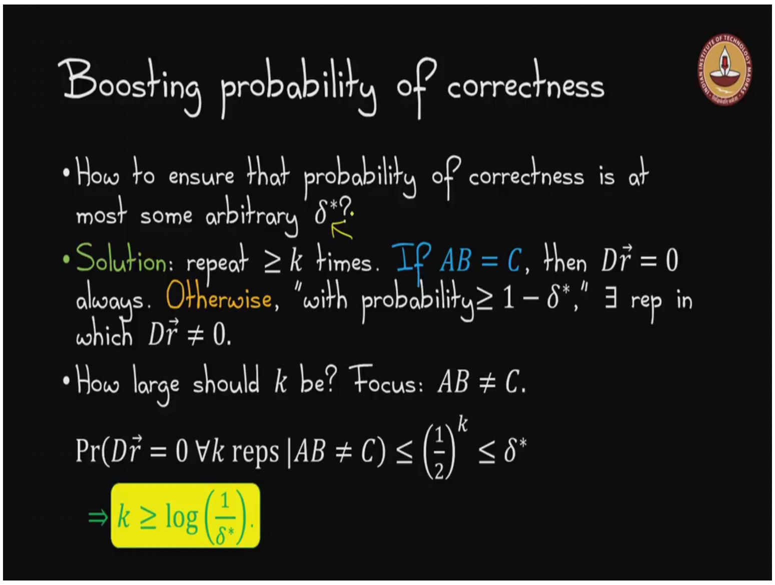
Is $\delta = \frac{1}{2}$ good enough?

NO!

So, what we have shown is that delta the up upper bound on the probability of the bad event delta is half is that good enough certainly not, I mean you would not bet your life on something where the probability of the bad event is close to half.

So, how do we work with that one thing that we can take advantage of is the fact that this is a problem where we have in an algorithm where it has one sided error ok, when A B is equal to C were always going to be correct when A B is not equal to C we are going to be wrong with probability at most half.

(Refer Slide Time: 21:05)



Boosting probability of correctness

- How to ensure that probability of correctness is at most some arbitrary δ^*?
- Solution: repeat $\geq k$ times. If $AB = C$, then $D\vec{r} = 0$ always. Otherwise, "with probability$\geq 1 - \delta^*$," $\exists$ rep in which $D\vec{r} \neq 0$.
- How large should k be? Focus: $AB \neq C$.

$$\Pr(D\vec{r} = 0 \;\forall k \text{ reps} \mid AB \neq C) \leq \left(\frac{1}{2}\right)^k \leq \delta^*$$

$$\Rightarrow \boxed{k \geq \log\left(\frac{1}{\delta^*}\right)}.$$

So, what we want to do now basically ensure that the probability of error is it comes down to some arbitrary delta star. So, you decide what the delta star is and the delta star could be like let us say 0. 0 0 0 1 ok, you decide what that delta star is going to be.

Now, what we are going to do is try to repeat this algorithm some k times and ensure that we bring down the probability of error to at most this delta star that your favorite delta star. So, of course, it is being in the one side of the thing when A B is equal to C we are always going to be correct. So, we are done with that. So, we are only going to worry about the case where A B is not equal to C.

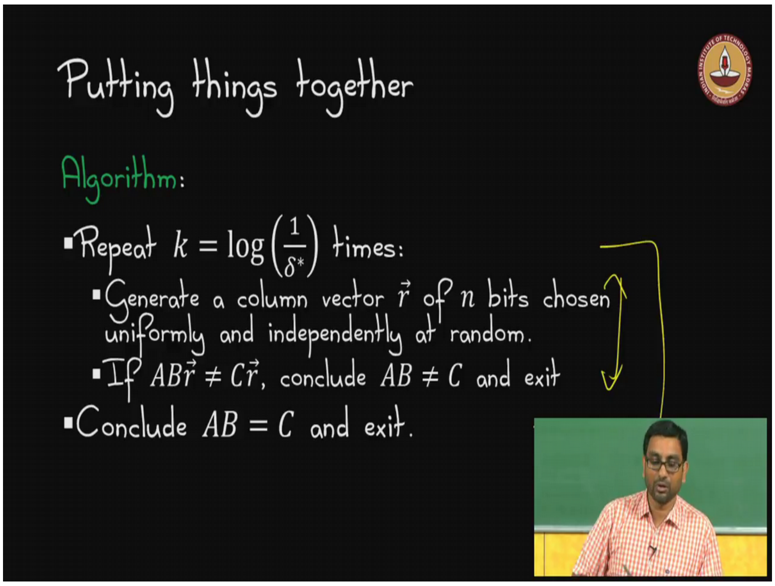
So, what is the probability that this bad even D r equal to 0 is going to happen for all the k repetitions given that A B is not equal to C ok, that is the question and remember these repetitions are going to be independent repetitions.

So, you can simply if they are independent you can just multiply them. So, it is going to be half raised to the power k and we want this half raised to the power k to to be bounded by delta star. Which means that you have to run this for k greater than or equal to log of one over delta star number of times, if we run it this many times and remember log of one over delta star is actually a fairly small quantity it is basically log is just a representation of the number of bits needed to represent A quantity right.

One over delta star could be something like if it is your delta star is say 0.0 0 1 your your 1 over delta star is like 1000 log of 1000 is what something like 1500 24 log of thousand 24 is A 10.

So, that is all if you just repeat it 10 times you bring down the probability of error down to your favorite delta star.

(Refer Slide Time: 23:13)



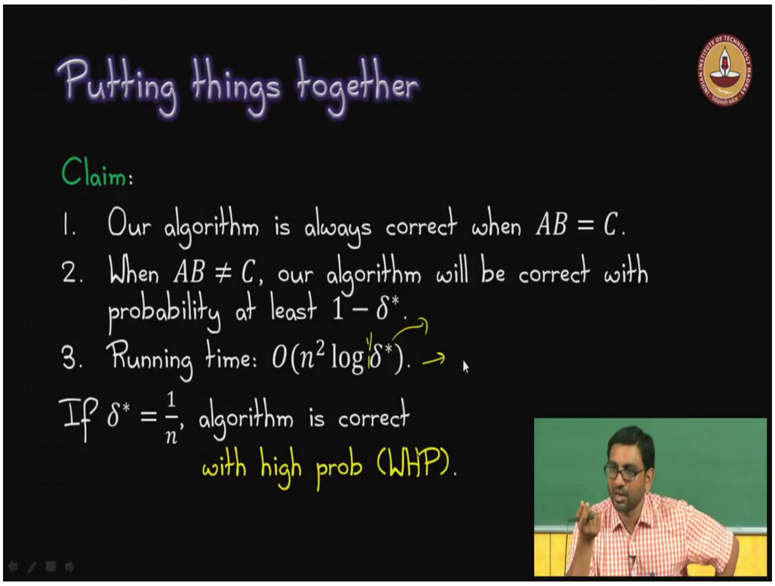
Putting things together

Algorithm:

- Repeat $k = \log\left(\frac{1}{\delta^*}\right)$ times:
  - Generate a column vector $\vec{r}$ of n bits chosen uniformly and independently at random.
  - If $AB\vec{r} \neq C\vec{r}$, conclude $AB \neq C$ and exit
- Conclude $AB = C$ and exit.

So, putting things together so, house just to this is the algorithm that we have already seen and were just were just gonna have to wrap it around the for loop and that is it.

(Refer Slide Time: 23:27)



Putting things together

Claim:

1. Our algorithm is always correct when $AB = C$.
2. When $AB \neq C$, our algorithm will be correct with probability at least $1 - \delta^*$.
3. Running time: $O(n^2 \log {}^{1}\!/\delta^*)$.

If $\delta^* = \frac{1}{n}$, algorithm is correct with high prob (WHP).

So, just remind ourselves what the claim is or algorithm is always correct when A B equal to C when A B is not equal to C our algorithm will be correct with probability at least one minus delta star.

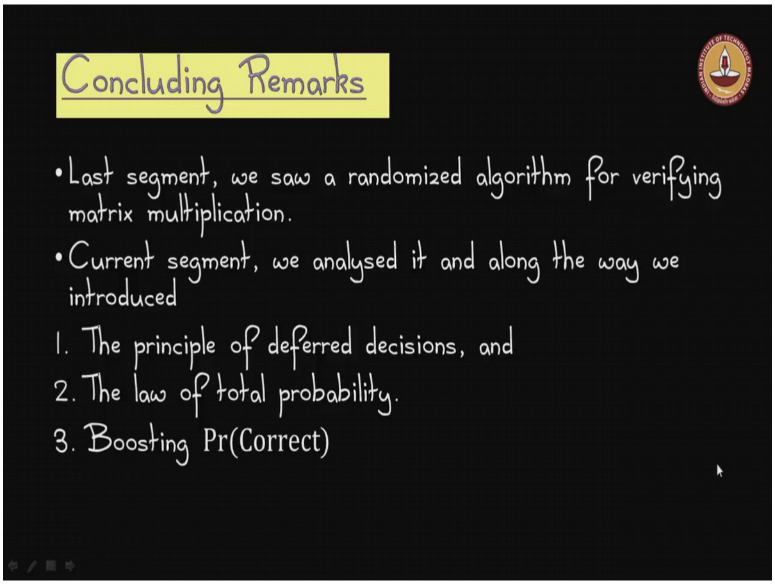
So, let us remember it is going to be incorrect with probability at most delta star. So, it is going to be correct with at least 1 minus delta star probability. And what is the running time well we know that the original algorithm was n squared running time, we are wrapping it around the for loop that takes log delta star number of iterations and such n squared log delta squared.

Now, one way to think about this delta star is you try to you you we in comparisons we are often interested in scalability the larger the problem we want we want to ensure that our guarantees are strong right. So, what we let us say we want our delta star to be equal to 1 over n ok. And when we get a correctness of this form where the in correctness probability is at most 1 over n and therefore, the correctness is at least 1 minus 1 over n we say that the algorithm is correct with high probability this is a standard term used these things.

So, when we get the probability of correctness to be at least 1 minus 1 over n, let us correct with high probability ok. And how do we ensure that we can get with high probability? And that is that is easy right. So, now, 1 over delta star; so, this running time if we want high probability this running time will just become basically n squared log n ok.

So, if you just add a factor of log n you are going to get with high probability correctness. So, just we are down to the concluding slide.

(Refer Slide Time: 25:15)



Concluding Remarks

- Last segment, we saw a randomized algorithm for verifying matrix multiplication.
- Current segment, we analysed it and along the way we introduced

1. The principle of deferred decisions, and
2. The law of total probability.
3. Boosting Pr(Correct)

So, what we saw? So, let us just conclude our segment in the previous segment, we saw the algorithm to verify matrix multiplication what we have done is carefully go through the analysis of this algorithm, we have shown that we will studied principle of deferred decision the law of total probability and we shown that it is correct with high probability.

So, with that let us look forward to the next segment, it is going to be another exciting algorithm called Kargers Mincut algorithm.

Thank you.