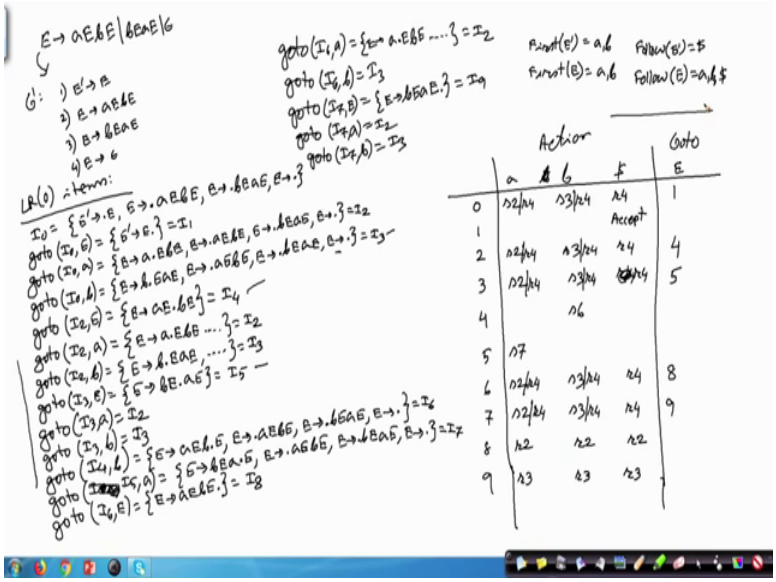
Compiler Design
Prof. Santanu Chattopadhyay
Department of E & EC Engineering
Indian Institute of Technology, Kharagpur

Lecture – 36
Parser (Contd.)

(Refer Slide Time: 00:14)



So, next, we have to make the remaining gotos. So, this goto I 6, a we have done so, I 6, a goto I 6, a, so that will be giving me the item E producing a dot E b E. Now, a dot b E a dot EbE so, that is same as this item I 2, so, etcetera. So, that is same as I 2. So, I 6, a is I 2 then I 6, b I 6, b. I 6, b will give us this b dot etcetera b dot E etcetera and that is nothing, but I 3. So, that is the set I 3. So, this will give me I 3 only. Now, goto I 6 something more is there dot a dot b is done. So, E is also done. So, I 6 is over.

I 7, E; I 7, E will give me this b E b E. So, I 7 E goto I 7, E will give us E producing bE aE dot and that is a new set. So, this is I 9. Now, I 7, E I have done I 7 will give me a dot something. So, goto I 7, a is a dot something that is I 2 and goto I 7, b will be b dot something. So, b dot something is I 3. So, this is I 3. Now, I 7 then what about I 8? So, I 8 there is no go because dot is at the end I 9 also dot is at the end. So, no new set can come from there. So, that completes the construction of this LR 0 items.

After that we have to look for the first and follow of this non-terminals. So, first of E dash first of E. So, let us look into these two sets. So, first of E is definitely a and b, ok.

So, this is first of E is a and b and first of E dash is equal to first of E, so, that is also equal to a and b. And, now follow of E dash and follow of E, ok. So, follow of E you can see here a and b is followed by a. So, this is a and b and since E dash is the start symbol of the grammar. So, this has got dollar in it and by the first rule. So, whatever is in follow of E dash is in follow of E. So, dollar is also there in this set. So, this is a b dollar and that is dollar.

After this we try to construct the parsing table for this particular grammar ok. So, SLR parsing table; this will have this action part and the goto part. So, I have got the symbols a, b and dollar. These are the three symbols you know terminal symbols and non-terminal I have got only E and I have got the states ranging from 0 to 9, ok. So, this is 0, 1, 2, 3, 4, 5, 6, 7, 8, 9; I have got nine such cases. Now, I have to look into individual items and see what can be done with that.

Now, looking at I 0; so, I 0, a is a shift operation and I 0, a is 2. So, this is shift 2. I 0, b; I 0, b is 3. So, this is shift 3 and then in I 0 I have got E producing dot so, I have to look into the follow of E and there I have to add the reduce rule E producing; so, E producing epsilon. So, this is reduce by rule number 4. So, this is this has got another entry now reduce by rule number 4, this has got another entry now reduce by rule number 4. And, this 0, E is 1, so, I have to make it 1.

Now, I 0 is over now I 1. So, E dash producing E dot. So, then this is E dash follow has got dollar. So, this should be accept this should be accept. Now comes this I 2 and in I 2 I have got this one a dot a and then I 2, a is I 2 only. So, so this is shift 2 then I 2, b; I 2, b is I 3. So, this is shift 3 and then I have got this rule E dot E dot. So, the follow set I have to see. The follow set of E is a, b and dollar there I will have to do reduction by rule number 4. So, all these are to be added reduce by rule number 4, reduce by rule number 4 reduce by rule number 4. So, they are to be added and I 2, E; I 2, E is 4. So, this should be 4.

Now, comes to now we come to this set I 3 and I 3 has got E a here. So, this I 3, a; I 3, a is I 2. So, this should be shift 2 from this item I can say it is shift I 3, b; I 3, b is 3 only. So, this is shift 3. Now, this item tells me that I have to add the reduce rules. So, there this reduce rules are to be added. So, all these reduced rules are added sorry this is reduce by rule number 4 and this I 3, E is I 5. So, this should be 5. Now, for item I 4 there is

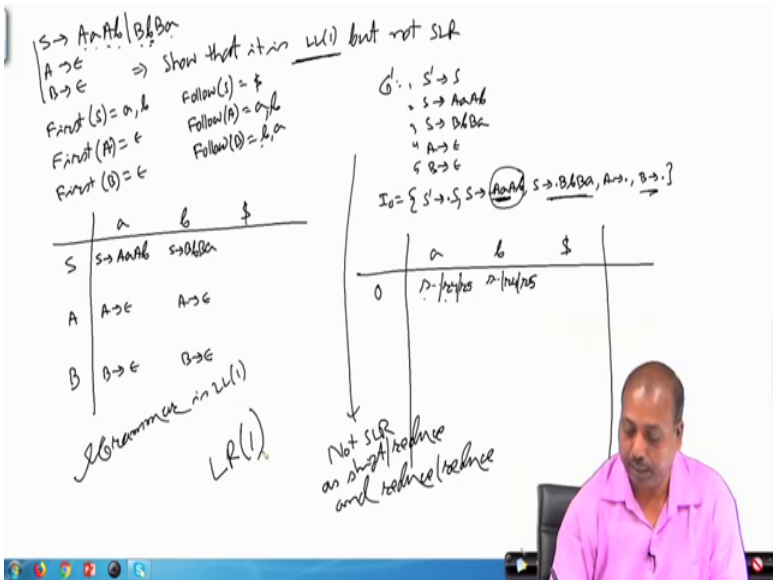
only shift action on b and I 4, b is I 6. So, I 4, b is shift 6. So, I 4 is done now comes to I 5. So, I 5 is this a. So, I 5 a is I 7. So, this is shift 7. So, there is no conflict at this point. So, I 5 is done.

Now, come to I 6. So, I 6, a is a shift operation and I 6, a is I 2. So, this is shift 2. I 6, b is I 3, so, that is shift 3 and then by this E producing dot. So, all this reduction by rule number 4 will come, reduce by rule number 4, reduce by rule number 4. And, your I 6, E I 6, E is I 8; so, this goto part it will have an entry 8. Now, come to item I 7 in I 7 I have got this one a dot a. So, this should be shift and I 7, a is 2. So, this is shift 2 then I 7, b; I 7, b is 3. So, this is shift 3 and this E producing dot. So, this follow of E a, b dollar I have to add the reduce rule reduce by rule number 4. So, they are all reduce by rule number 4 and I 7, E I 7, E is what I 7, E is I 9. So, this is 9.

Now, come to item 8, ok. So, the set 8 I 8. So, there I have got a rule this thing that is rule number 2. So, I have to look into the follow of E that is a, b dollar and there I have to reduce by rule number 2. So, it says it is reduce by rule number 2, reduce by rule number 2, reduce by rule number 2. And, then for I 9; for I 9 I will have this rule. So, that is rule number 3 and so, all of them should be reduce by rule number 3 fine.

So, this way we can construct the SLR parsing table for this grammar. So, essentially you have to patiently construct this LR 0 items, then you have to patiently construct this follow sets and rest of the thing is a bit mechanical. So, you have to look into this individual item sets and see whether you can get a you can get a either shift or reduce for different actions. So, that can be done, ok.

(Refer Slide Time: 11:01)



Next, we will be looking into a grammar which is given by the expression which is ok. So, next we will be looking into a grammar which is very simple, but we will see that it is not always mandatory that a grammar has to be SLR and there so, if we look into the corresponding LL grammar. So, maybe the normally it is seen that if a grammar is LL so, it is also a SLR, but there are some exceptions also. So, next we will be looking into a grammar where we have got that type of exception.

So, the grammar is given by this; so, s producing S a sorry Aa Ab or BbBa and then A producing epsilon and B producing epsilon. So, this is a grammar and that is given to us and the question is to show that it is SLR, it is LL 1, but not SLR. So for solving this problem, so, first we try to show that this is LL 1. For showing LL 1, so, we have to construct the first and follow sets.

So, we first construct the first of S first of A and first of B ok. So, first of S is equal to first of A and first of B and first of A has only epsilon in it. So, this has got only epsilon in it and then since A can be reduced to epsilon. So, this set this first contains A and from this B can be reduced to epsilon, so, this can contain B. So, that is the first set.

And, the follow set; so, follow of S then follow of A and follow of B ok. So, follow of S since S is the start symbol dollar is there, then it says that the follow of A, so, A can be followed by small a. So, this is small a and A can be followed by b. So, this is also there

a and b. Similarly follow of B, so, B may be followed by B small b and B may be followed by small a. So, that is also same b or a.

So, after constructing this follow sets so, we first and follow sets we try to construct the predictive parsing table. So, if we can successfully construct this table then the grammar is definitely LL 1. So, we have got S, A and B and we have got the terminals a, b and dollar. Now, coming to the first rule S producing AaAb; so, this so, you have to looking to the first of this and first of this whole thing contains this small a because A can be reduced to epsilon. So, this rule is added here S producing AaAb. And, for the second rule it says that I can since this can the first set can contain the small b so, this will be added Bb Ba,.

Then, this A producing epsilon; so, I have to look into the follow set of A and add this rule here. So, A producing epsilon; A producing epsilon and for the third rule B producing epsilon, so, I have to add the rules there B producing epsilon, B producing epsilon. So, for this grammar you see that there is no entry in this parsing table which is multiply defined. So, this grammar is LL 1. So, the grammar is LL 1.

Now, what about the SLR case? So, for constructing the SLR parsing table, so, we first augment the grammar. So, we get the augmented grammar G dash as S dash producing S, S producing AaAb, S producing BbBa and A producing epsilon, B producing epsilon. After this we have to set the LR 0 item. So, first so, I 0 is constructed as S dash producing dot S and then S produces dot AaAb, S produces dot BbBa, A produces dot B produces dot. So, this is the I 0, ok.

Now, we can construct the full set of items, but looking at this one itself we can try to understand like what will be the action part. So, this is my set I 0; then these are the terminal symbols a, b and dollar in the action part. Now, this rule this item tells me that whatever is in first of A, so, there I have to whatever is in first of this right hand side so, there I have to do a shift operation. So, since this first of this right hand side A can be reduced to epsilon, so, first of this contain this small a. So, this says that shift by something a shift I am going to a new state. Similarly, this item will tell me that you do a shift and go to a new state.

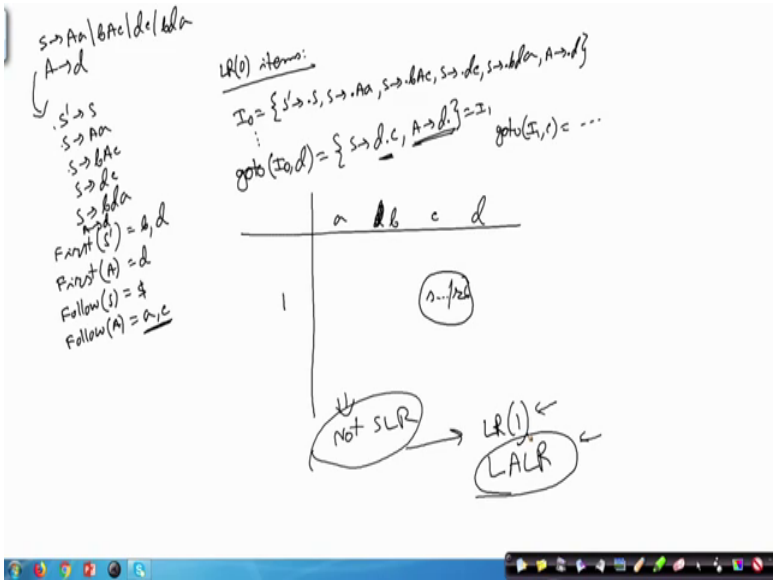
Now, this A producing dot is there, so, that means, that it tells me to do a reduction; reduction for all the symbols which are in the follow of A. So, follow of A contains a and

b. So, this will also suggest to do a reduction by rule number say 1, 2, 3, 4 5 reduction by rule number 4. This will also tell that you do a reduction by rule number 4. Now, when we come to this item B produces dot, so, that says that whatever is in follow of B there I have to do a reduction by this particular rule. Follow of B contains b and a; so, this will have reduce by rule number 5 this will also have reduce by rule number 5.

So, in this item itself you see that we have got shift-reduce conflict as well as reduce-reduce conflict. So, this grammar that we are trying, so, this is not SLR. So, this is not SLR as both shift-reduce and reduce-reduce conflicts are coming shift-reduce and reduce-reduce conflicts are coming. So, that way we can find some cases where the grammar is not SLR, but the grammar is LL 1. Of course, you can you may see that this grammar can be we for this grammar we can make say LR LR 1 parser. So, if we are doing this canonical LR it may be possible that we do not get this type of conflicts. So, that is that we will see later.

So, that way whatever the statement that wherever you can get and LL parser so, we can also get an LR parser. So, that statement is not violated by this example. This example just shows that there may be the case where the grammar is LL 1, but not SLR. So, it may be the grammar is LR, but it is not SLR. So, only that part ok. So, this way you can you have to see like what is happening to the grammar by looking into the corresponding table and then we have to look into the corresponding sets.

(Refer Slide Time: 20:05)



Next, we consider another example which is where we try to show that the grammar is not SLR. Say we have got a grammar like this say S producing Aa or bAc or dc or this bda where the S and A they are the non terminals of this grammar and rest are all terminal symbols and A produces d. So, this is the grammar that we have.

So, to see that whether it is an SLR or not first of all we do this augmentation S dash producing S S producing Aa, S producing bAc, S producing dc, S producing bda, fine. The next part of the job is to find out the first and follow sets. So, the first of S dash; so, since we have to show that whether it is SLR or not so, we first do we do this first follow computation before going to a LR 0 item because that may help us in identifying the negative cases much earlier.

So, first of A is equal to d definitely because that can only give me this thing and first of S is b and d, now that is the first. Now, the follow set follow of S and this follow of A; so, follow of a you can see a small a can come, small c can come ok. So, this is follow of A and follow of S since it is the start symbol so, this is dollar ok. So, this is the follow set.

Now, if you try to construct this SLR the LR 0 items. So, now, we try to construct the LR 0 items. Now, I 0 is S dash producing dot S comma S producing dot Aa, S producing dot bAc, S producing dot dc and S producing dot bda. Oh, that other rule is there that A producing d, so, that is not written here and a producing dot d. So, this is the LR 0 item I 0.

Now, I have to construct this gotos. So, you can construct all those gotos, but let us see what happens to this goto I 0, d? goto I 0, d will tell me that this S producing d dot c will be one item and A producing d dot will be another item fine. So, if this item I call I 1, then when I am trying to construct the SLR parsing table so, for this cd etcetera. So, we have got the terminals a, b, c, d.

Now, I have got this say this set is I 1. Now, for I 1, when I am trying to look into this individual items so, this rule will tell me that you should do a shift operation. So, this 1, c should be a shift to some state fine. So, whatever be the goto I 1, c. So, whatever be the goto I 1, c equal to so, to that state it will go.

Now, what about this rule this item, so A producing d dot? So, it says that you have to look into the follow of A and there you have to do the deduction. So, follow of A is a and c. So, you can understand that this will give me a reduce it will give me a reduce by rule number whatever be the rule number. So, a so, it is 1, 2, 3, 4, 5, 6 it will give me a reduce by rule number 6. So, this way it gives rise to a shift reduce conflict here. So, this will give rise to a shift reduce conflict here and the grammar will not be an SLR grammar. So, this is not SLR because it is giving a conflict like this.

Now, it can be shown that this can successfully give me LR 1 parser it can successfully give me LR 1 parser. In fact, for this particular grammar you can see that this can also give me LALR parser, ok. So, we will see the, we will do that exercise later we will see that this particular grammar though it is not SLR. So, it is LR 1 as well as it is LALR 1, but it is not SLR, ok.

So, we will continue in the next class.