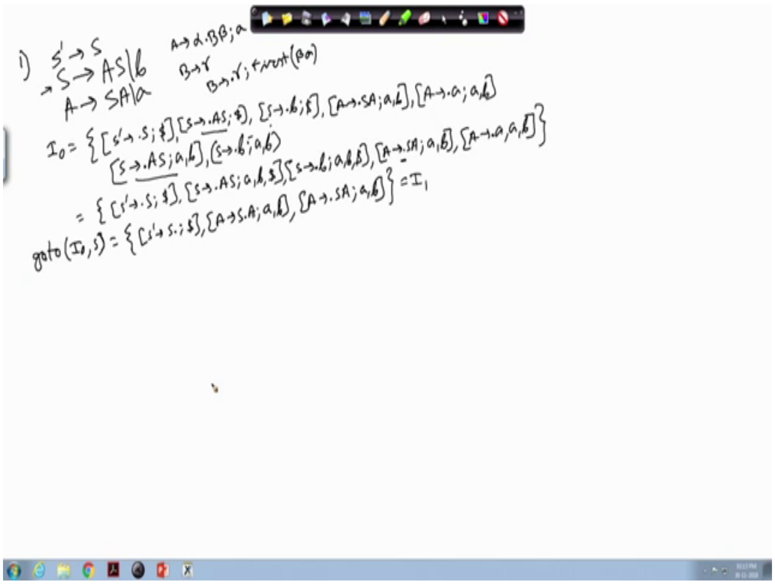
Compiler Design
Prof. Santanu Chattopadhyay
Department of E & EC Engineering
Indian Institute of Technology, Kharagpur

Lecture - 37
Parser (Contd.)

So, today we will look into some more examples of these parsing techniques particularly the LR parsers and LALR parsers. In the previous classes we have seen examples of SLR parsing. So, today we will first look into some LR parsing at a table construction method.

(Refer Slide Time: 00:33)



1) $S' \to S$
$S \to AS \mid b$
$A \to SA \mid a$

$A \to \alpha.B\beta; a$
$B \to \gamma$
$B \to .\gamma; first(\beta a)$

$I_0 = \{[S' \to .S; \$], [S \to .AS; \$), [S \to .b; \$], [A \to .SA; a,b], [A \to .a; a,b]$
$[S \to .AS; a,b], (S \to .b; a,b)$

$= \{[S' \to .S; \$], [S \to .AS; a,b,\$], [S \to .b; a,b,\$], [A \to .SA; a,b], [A \to .a, a, b]\}$

$goto(I_0, S) = \{[S' \to S.; \$], [A \to S.A; a,b], [A \to .SA; a,b]\} = I_1$

So, the grammar that we will consider is it will be a simple one and so that we have got less number of states, but you will see that even with that less number of states. So, that is going to take quite some number of states compared to this SLR even if the grammar is simple. So, this LR parsers it will have large number of states.

So, the grammar that we consider is like this we have got this S producing AS or b and then we have got A producing SA or a. So, this is the grammar. So, you have got the where this S capital A so these are non-terminal symbols and small a and small b these are terminal symbols. So, the first step is to augment the grammar by putting this S dash just producing S as one production and then we have to construct the set I 0. So, I 0 is the closer of the item this first production S dash producing dot S with look ahead dollar and then we have to consider the closer of this and closer of this will have all these things like S producing dot AS dollar then S producing dot b and dollar.

So, like that will have two rules and after that, since we have got this S producing AS and this S dot a is there. So, I have to construct or to see what is A producing things. So, in that way I will get the items like A producing dot SA and then the rule says that when you are doing this S producing A producing dot SA then, I have to say what is what will be the terminal symbols after this a look ahead symbols. And, the look ahead symbols rule was like this that; if you have got a rule like A producing alpha dot B beta then and if there is a rule like B producing gamma then I will add this item B producing dot gamma to this set and the look ahead will be the first of so, there is a.

So, this was the item alpha dot B beta semicolon a. So, a was the look ahead. So, I have to look into the look ahead of first of beta a. So, whatever comes in the first of beta a so, all those symbols will come here. Now, you see that when I am looking into this particular production this particular item S dash S producing dot AS then, I have got these things. So, dot is before a so, the beta part is capital S. So, I have to see what is the first of capital S and first of capital S you see b will come and a will come. So, while adding this item the look ahead symbols will have both a and b. So, that way the look ahead symbols will be added and then there is another rule a producing dot a.

So, that will also come A producing dot a and the look ahead symbols a and b. So, they will come. So, this way this item will get added and now you have got these things this A producing dot SA. So, A producing dot SA so, again I have to apply this rule. So, this dot S. So, this from this I will get the item S producing dot AS and then I have to see whatever is in the first of A. So, first of A can have this small a as well as this small b both of them can be there in the first of A. So, that that way the look ahead it will be a and b ok. So, this is another item added.

Now, this thing we can simplify now like now you see that this item and this item they are core part is same only the look ahead part is different. So, this is a b and this is dollar. So, we will write it together. So, ultimately we will get it the items like this. So, S dash producing dot S semi colon dollar this will be there then S producing dot AS and the items the look aheads will be a b and dollar ok.

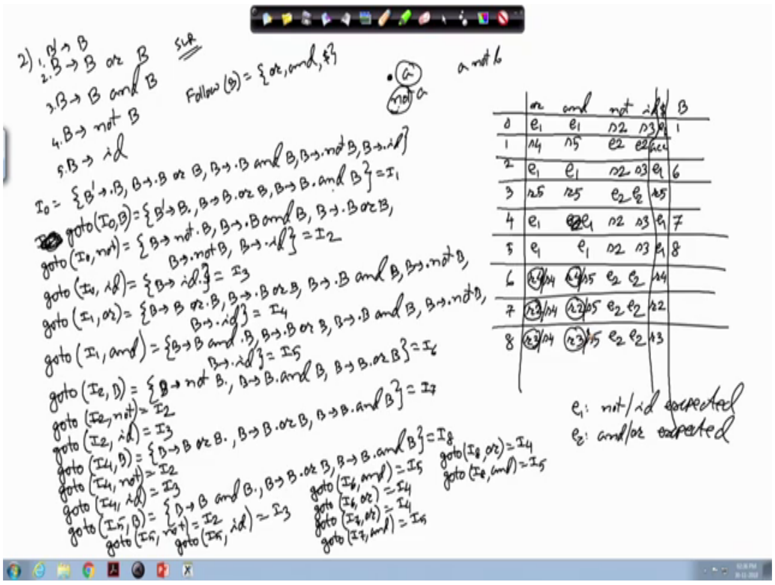
Now, I will have this S producing dot b. So, S producing dot b and here also I will get the look ahead a and b. So, like just is in this case we had this. So, for this one also I will have this a, b and dollar so, a b. So, this item will also be there. So, it is not I have not written it. So, this S producing dot b will come with the look aheads a and b. So, this item is also there now I can club these two because their core part is same. So, this is a b dollar and then I have got this a producing dot SA a, b then A producing dot a a b.

So, this is the item setup item I 0. Now for I 1 so, I have to do go to I 0, S; go to I 0 S will give me from the first one I will get S dash producing S dot semi colon dollar. So, this is giving me then dot S. So, this rule is there so, this will give me A producing S dot A semi colon a, b then since this S producing dot A is there. So, I have to take this A producing things also. So, I will get the items A producing dot SA with the look aheads of first of A. So, first of A has got both a and b. So, a and b is already there so, I will get this A producing dot SA with a, b ok.

So, then there is this A producing dot SA is there. So, this will give us A producing dot A producing S dot a. So, that is there and this will be the thing like I will have this production then A pro A this there is A dot before a. So, this somewhere I 0 S. So, this S producing this one S producing S dot A and then I will have this from this rule I will get S producing S dot A ok. So, this will be the set I 1 ok. So, this way we have to go on constructing the different items and there will be a large number of items created. So, it is very difficult to do it by hand.

So, but in this way if you continue then you will get the set of items that are coming and then you can combine them using some; if some of them are same then you can combine some of the items are same you can combine them. And, then ultimately you can make the parsing table and from the parsing table you can go into the parsing part of it. So, this way we can construct the table. So, next we will be looking into another example where we try to we will be doing SLR parsing only, but we will try to see like how this error functions are taken care of. How this so, for that purpose so, we take the grammar.

(Refer Slide Time: 19:29)



2) 1. B' → B
2. B → B or B
3. B → B and B
4. B → not B
5. B → id

SLR

Follow (B) = {or, and, $}

.(a) (not) a   a not b

I0 = {B' → .B, B → .B or B, B → .B and B, B → .not B, B → .id}

goto(I0, B) = {B' → B., B → B. or B, B → B. and B} = I1

goto(I0, not) = {B → not .B, B → .B and B, B → .B or B, B → .not B, B → .id} = I2

goto (I0, id) = {B → id.} = I3

goto (I1, or) = {B → B or .B, B → .B or B, B → .B and B, B → .not B, B → .id} = I4

goto (I1, and) = {B → B and .B, B → .B or B, B → .B and B, B → .not B, B → .id} = I5

goto (I2, B) = {B → not B., B → B. and B, B → B. or B} = I6

goto (I2, not) = I2

goto (I2, id) = I3

goto (I4, B) = {B → B or B., B → B. or B, B → B. and B} = I7

goto (I4, not) = I2

goto (I4, id) = I3

goto (I5, B) = {B → B and B., B → B. or B, B → B. and B} = I8

goto (I5, not) = I2

goto (I5, id) = I3

goto (I6, and) = I5

goto (I6, or) = I4

goto (I7, or) = I4

goto (I7, and) = I5

goto (I8, or) = I4

goto (I8, and) = I5

	or	and	not	id	$	B
0	e1	e1	s2	s3	e1	1
1	s4	s5	e2	e2	acc	
2	e1	e1	s2	s3	e1	6
3	r5	r5	e2	e2	r5	
4	e1	e1	s2	s3	e1	7
5	e1	e1	s2	s3	e1	8
6	r4/s4	r4/s5	e2	e2	r4	
7	r2/s4	r2/s5	e2	e2	r2	
8	r3/s4	r3/s5	e2	e2	r3	

e1: not/id expected
e2: and/or expected

We take the grammar which is for the Boolean expressions. So, Boolean expression grammar is like this. So, B producing B or B, then B producing B and B, then B producing not of B and B producing id. So, if this is the grammar then we have to add that initial production B dash producing B. So, these are the rules that we have. So, these rules are number like 1, 2, this is 3, this is 4, this is 5. Now, if we try to so, we are trying to construct an SLR parsing table and we will embed the error functions into it. So, what are the errors that will see and accordingly we will put the error productions also.

So, this LR 0 items so, the I 0 is given by this B dash producing dot B and then whatever it is so, all these rules will come B producing dot B or B, then B producing dot B and B, then B producing dot not of B and B producing dot id. So, this is the I 0 part. Now for the I 1 we will so, for so, we will see like go to I 0 B go to I 0 B so, this will give us B dash producing B dot, then B producing so, from here B producing B dot or B B producing B dot and B. So, these are the three items. So, this is the set I 1.

Now, I 0 the I 0 not is possible. So, go to I 0 not so, this will give us B producing not dot B and then all these rules will come that B producing dot B and B, B producing dot B or B, B producing dot not of B and B producing dot id. So, all these rules will come. So, this is my set I 2. Now go to I 0 id; go to I 0 id is B producing id dot. So, this is the set I 3. Now for from I 1, I can get go to I 1 or this will give me B producing B or dot B and since dot B is there. So, all these items will come B producing dot B or B, B producing dot B and B, B producing dot not of B and B producing dot id. So, that will be the set I 4.

Now, go to I 1 and go to I 1 and so, this will give us B producing B and dot B and then all these rules will come. So, B producing dot B or B, B producing dot B and B, B producing dot not of B and B producing dot id. So, this is my I 5. Now, with I 1 everything is done, now have to do with I 2 ok. So, then this one is there B. So, go to I 2 B. So, this is B producing not of B dot and then this one B producing B dot and B then, B producing B dot or B ok. So, that will give us set I 6. Then I 2 B I have done, then I 2 not go to I 2 not; go to I 2 not so, this will give me B producing dot not dot B. So, that is same as that item I 2. So, go to I 2 not is I 2 only.

Then go to I 2 id, this is B producing id dot that is equal to I 3 fine. Now I 4 go to I as I from I 3 I cannot get anything from go to I 4 B. So, this will give me B producing B or B dot, then this one B producing B dot or B, then B producing B dot and B ok. So, these are the three production. So, that is my I 7 that is I 7. So, I 4 B is done. Now I 4; I 4 not go to I 4 not this will give me same as B dot B not dot B. So, B not dot B. So, that is the item I 2; so, this is I 2 only then go to I 4 id I 4 id is B producing id dot. So, B producing id dot is I 3. So, this is I 3 only. So, I 4 is over now I 5. So, go to I 5; go to I 5 B is B producing B and B dot and then this B producing B dot or B then, B producing B dot and B ok.

So, this is the item I 8. Now I 5 B is done. Now other possibilities from I 5 is I 5 not. So, go to I 5 not is not dot B. So, that is equal to I 2. So, that is equal to I 2 then go to I 5 id is B producing id dot. So, that is I 3 ok. So, I 5 is over. Now I 6 ok so, go to I 6 on and I 6 and will give me this item this one this I 5 only. So, this is B producing B and dot b. So, that is the from here I will get B and dot b. So, it will give me I 5. So, go to I 6 and this I 5 go to I 7. So, I 6 is go to I 6 or go to I 6 or is B or dot. So, that is I 4 ok. So, I 6 is over.

Now, from I 7 I will get go to I 7 or so, this will give me B or dot B so, that is I 4 and I 7 and I 7 and will give me B and dot b. So, that is I 5 and from this set I 7 so, I 7, I have done from I 8. So, go to I 8 or will give me B or dot B. So, that is I 4 and this one go to I 8 and will give me B and dot B. So, B and dot B that is I 5 ok. So, this finishes construction of these items sets of items now we can construct the parsing table for so, for doing the for our purpose the parsing table construction. So, I have got 9 states. So, the states are the items are like 0, 1, 2, 3, 4, 5, 6, 7 and 8 and that terminal symbols that I have or, and, not and id.

And then the go to part I have since I have got a single non-terminal B. So, that is B and also I need to constitute the follow set; the follow set of B; follow set of B it has got or, it has got and ok. So, follow set of B has got or and. And of course, since we have got B as the start symbol of the grammar. So, dollar will also be there B or dollar or and dollar. Now I have to look into the set I 0. Now, it says that this B producing dot not B is there. So, I have to I have to on not I 0 not is I 2. So, this will B shift 2 ok.

Then I 0 id I 0 id is I 3 I 0 id is I 3. So, this will be shift 3. Now these two entries or and. So, they are blank. So, they are actually error entries and I 0 B; I 0 B is I 0 B is I 1. So, this is 1 now from I 1. So, I 1 there is a B dash producing B dot. So, I will have this one as this dollar is I have not written that dollar. So, dollar is accept here; the dollar is accept and then you see that we have got this or dot before or. So, I 1 or is I 4. So, this is shift 4 and then, and is shift 5 sorry I am writing at a wrong place. So, this is I 1 so, shift 4 and shift 5 and this dollar is this one then the part before that is id.

Now, this from I 2 from and there is from I 2 you have to see like there is a dot before a terminal symbol at not. So, I 2 not I 2 not is I 2. So, this is shift 2 and I 2 id; I 2 id is I 3. So, this is shift 3. So, these two are done there is no reduction possible in I 2. Now I 3 it is a B producing id dot. So, this is telling me in the follow of B or, and and dollars. So, I have to make it this one. So, this rule number 5. So, or and in I 3 so, this is reduced by rule number 5; this is also reduced by rule number 5 and this dollar is also reduced by rule number 5 ok.

So, this is I 3 is done. Now come to I 4; in I 4 I have got dot before this not and I 4 not is I 2. So, this is shift 2 and I 4 id is; I 4 id is I 3. So, this is shift 3 and in I 4 I do not have any reduction because there is no dot at the end. So, that is done. Now come to I 5 then I 5 then there is 2, I 2 there was go to B or something I 2 B is I 6. So, this will be 6 that is there. Now come to the set I 4 I have done. Now I 5, I 5 again there is a dot before not and dot before id and I 5 not is I 2; I 5 not is shift 2 and I 5 id is I 5 id is shift 3. So, this two remain as it is and then I have this I 5 B; I 5 B is I 8. So, I 5 B is I 8. So, that will be there.

And what is I 4 B; I 4 B is I 7. So, this will be 7 fine. So, I 5 I do not have any dot at the end so, nothing more to do. Now I 6, I 6 I have got a dot at the end. So, for the follow set I have to do reduce by rule number 4. So, or and dollar. So, there will be reduced by rule number 4, reduced by rule number 4, reduced by rule number 4 fine I 6. Now that is for the first one. Now this one dot, and. So, this is telling me B dot B and B. Now this is I 6 and I 6 and is I 5. So, this also says that you shift and go to state 5 and then this I 6 or is I 4. So, this is also telling me to shift and go to 4 state 4.

Now, I 7 I 7 is here. So, I 7 in I 7 we see that this there is a B or dot B or B dot. So, this is the reduction will come. So, or and dollar. So, they will be reduced by rule number 2 or, and dollar they will reduce by rule number 2 and then I have got this thing this dot B or dot B dot or B. So, I 7 or is I 4. So, this will also say shift by go to state 4 and I 7 and is I 5. So, this will tell me shift by go to state 5. So, that will be finishing then this one I 7 I 7 I will have this situation. So, this is I 6 I 6, I 7 then I 8.

So, I 8 we have got this B producing and B dot. So, that is reduced by rule number 3 and here also I will get reduced by rule number 3, here also I will get reduced by rule number 3 and then in I 8, I have got this dot or. So, this will be this will tell me I 8 or is I 4. So, this is shift 4 and this is shift 5. So, that will be constituting the whole parsing table. Now after constituting the parsing table so, there are many entries which are blank fine. Now, I have to frame the error routines and accordingly I can put some error function there ok.

So, let us see, what is the expectation like at this point if you are so, you are at state 0 and then you are expecting that at this point you are got an or. So, your expression started with an or now how can a Boolean expression start? It can start with some identifier say a or it can start with the not symbol not a like that. So, either it has to start with not or an id. So, these two are the valid one. So, otherwise they are invalid. So, at this point I was expecting either a not or an id. So, accordingly I write here 1 error function e 1. So, for both of them this is e 1. So, here also this is e 1.

In state 1, when I am in state 1 then I have seen something with B and I am expecting or and. So, these are the two expectations. So, accordingly I can say as if I was expecting some operand ok. So, this instead of that I have got a not ok. So, it is the it has come like this some id a and then not B. So, that is that cannot be correct. So, this is also an error, but here I am expecting an operator the and or or and. So, this is either it is and or it is or.

So, these are the these another error function. So, we have got 2 error functions e 1 which is expecting an operand or in or the not and e 2 is we are expecting the operand or or and so, either of those two.

Similarly, in state 2 also so, you can see the I 2. So, it was expecting some to some operand or not. So, here also this is e 1 these two are e 1 similarly I 3 so, these two are e 2. So, this is also e 1 ok. So, this way you can fill it up like I 4, I 4 I was expecting again not or this thing id sorry this is also e 1 not or e 1, then in I 5 I was expecting in I 5 I was expecting to see some not or id. So, this is also e 1, in I 6 I was expecting and or or. So, these are e 2. This is also e 1, this is also e 1 and this is in I 7 I was expecting or and and. So, this is also operand expected in I 8 also I was expecting or an end. So, this is e 2.

So, where e 1 is not or id expected. So, this is the error function that we have and in e 2 we have got and or or expected. And of course, that is a shift reduce conflict at this point. So, here you are expecting the to end the expression, but you have got an or operator. So, instead of shifting so, I should do the reduction because that is a not id they have higher precedence then or. So, I will this so, this should be the rule. So, here also this should be the rule, here also I should do the reduction not shifting and here I will be doing this reduction only ok. So, that way you can incorporate this error functions into this parsing table and make the parsing table full so, that you can do error detection and recovery.