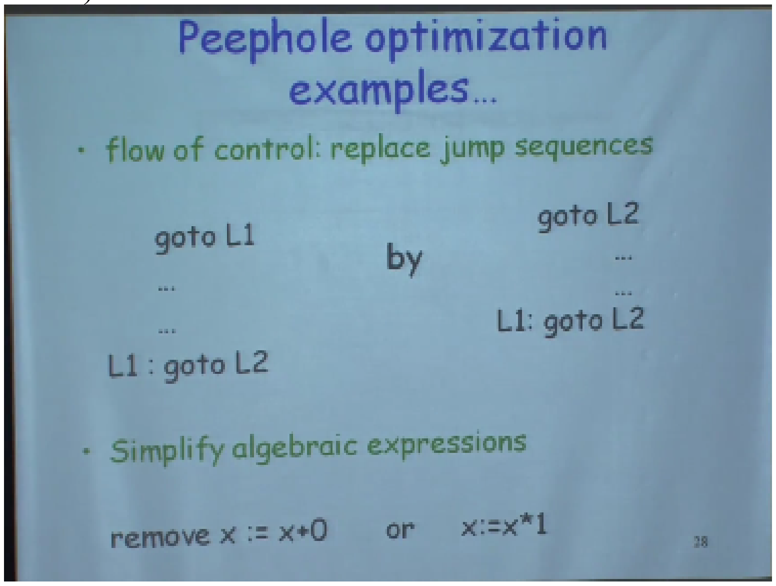
Indian Institute of Technology
Kanpur
NP – TEL
National Programme
On
Technology Enhanced Learning
Course Title
Compiler Design
Lecture – 30

by…
Prof. S. K. Aggarwal.
Dept. of Computer Science and Engineering.

Good morning so let us continue where we have previous class we start at the what kind of previous class whether we describe what is in the previous class what we are trying to do we have slighting description over the board and whenever you have find this slighting your size 4 and this result that is be operator windows whenever you find a shorter is the same operation the development.

The fist actually paper and we believes continuously went on to the work like saying why things physical window example people have branch description actual physically separately window which will have two instructions of branch the next description 0 to 3 part going to control so branch, so basically we are doing here is that we are looking at the target port.

(Refer Slide Time: 01:37)



Peephole optimization examples...

- flow of control: replace jump sequences

```
goto L1                 goto L2
...            by       ...
...                     ...
L1 : goto L2            L1: goto L2
```

- Simplify algebraic expressions

remove x := x+0 or x:=x*1

28

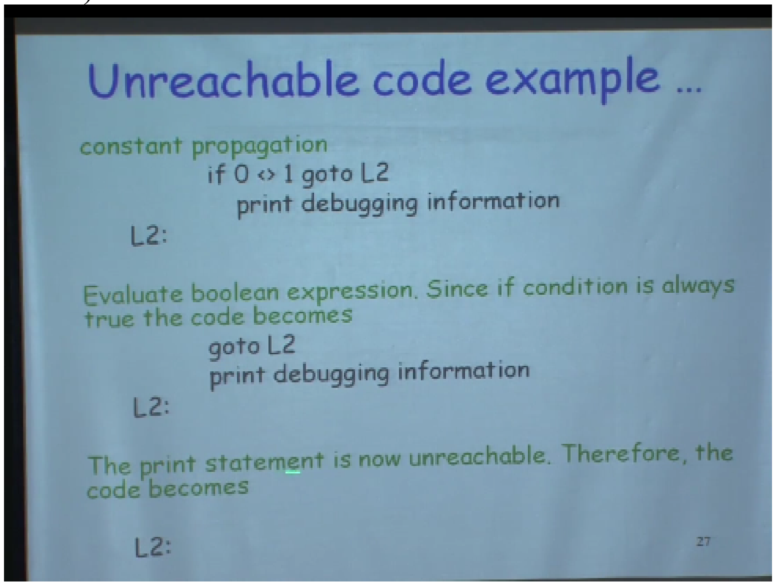
Which contains redundant suboptimal constructs we want to replace short sequence that is what a equivalence placed it by passed a sequence this is a small so let us look at what are time moving so what we have is for a example this is a first example we saw at suppose I have this athletic says removed codes and you see instructions and we know that irrespective of what the this can always unless have exclusive jump coming to the second remove okay, instruction always it will move this is one of my imagination that is possible okay.

Let us look at another example so suppose we have this sort sequence okay, and what we dealing with this unreachable force so suppose I have this we initiates zero and then I have conditional which says deeper we already seen this one can be may frequently use it may applications but if you recall me discuss some point of that you have the internally switchers and you can turn that on and off by just C and change this 0 to 1.

And then it looks complete information is going to finite statement machine and how does parser is going to a stack which a complete stack while parser so suppose I get into a situation like this and I know that this D-plug switch okay now can be done something may be interesting okay, now it says if debug what it actually means is something then it gets translated force something like if D-Plug is 1 and I will jump well 1 were I printing on the D plugging information.

Otherwise I am going to 2 normal excursion sequence and I can eliminate this jump over jump because this is a jump from here to here ,I can removed it by checking this condition and then over look something D-plug is not =1 then straight away go to 2 otherwise this information okay, so this how much we address code may look like not to try to optimize using optimization.

(Refer Slide Time: 04:15)



Unreachable code example ...

constant propagation

```
        if 0 <> 1 goto L2
            print debugging information
    L2:
```

Evaluate boolean expression. Since if condition is always true the code becomes

```
        goto L2
        print debugging information
    L2:
```

The print statement is now unreachable. Therefore, the code becomes

```
    L2:
```

27

What may happen that I will lose something called a constant most propagation says this we would not have time to time going to formal details of how the constant is propagation is done what it means that if I have variable this value is know at compared time and the value is constant then I can replace the variable why the value itself and then the after propagation I have at the condition like and say that 0 is not = .

Now I know that this Boolean expression can be evaluated at combine time and we know that definition wills obviously the condition will always the 0. =1 is false 0.1 will always condition wherever condition look at Boolean expression so this condition will always with true and therefore I can replace the whole thing by saying that the condition is jump okay all this elimination will get elimininted good and get moving window is when I say that executed statement is excitable and there go.

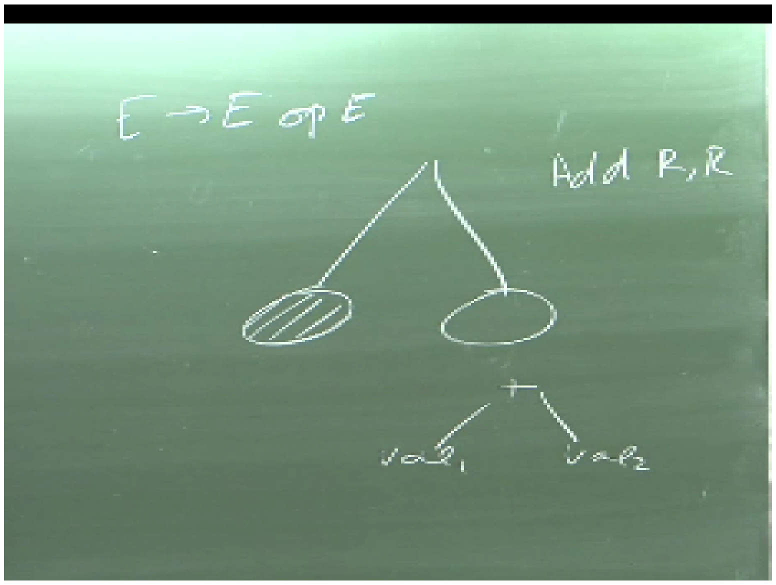
Because I may jump from places but defiantly this jump is redaction, because here jump if I eliminated this will fall this is one of optimization so such there many possible so I can flow that control and having sequence of jump so if I say go to L 1 and L 1 comes here then I know straight away that this place can be replaced by straight away while 2 and then this L 1 going to L2 can I eliminate this ,I cannot because possible that I am coming to L1 not only come here.

Logical rather than physical input okay and similarly when we talk about laphape expression in I can immediately see that these kind of expression so whenever I say that reduction is 0 participation is 1 I know that operation therefore I am just going to eliminate that similarly I can do what we know as stand deduction normally we will find that operators which will complete the same thing but some operators are going cause okay.

So for examples if I say that if I am trying to square exploitation normally a operation which is available in hardware I can have actually a function called but if I know that this is only square okay similarly if in say that if it is multiplication power of 2 or division by number of 2 okay, so these again our damp lets similarly another dimpled which may be possible is that if I say and 1 2 I can replace this two designed damp lets normally the compiler writers.

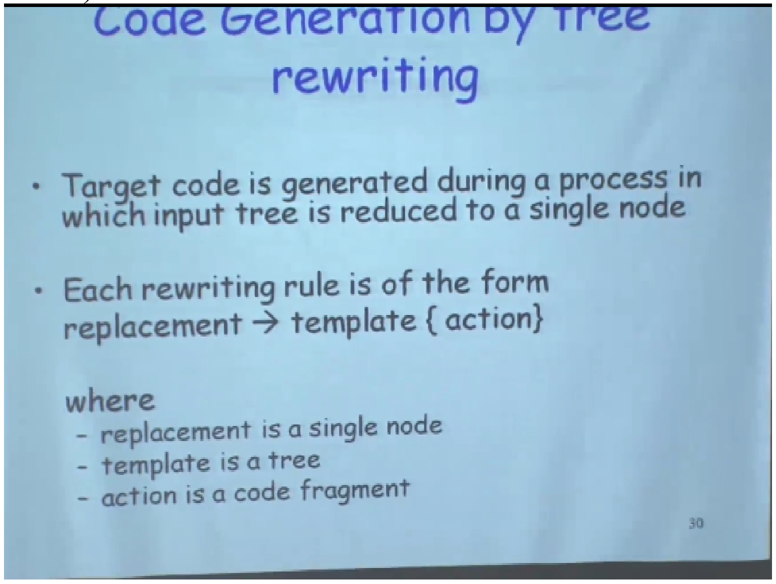
Who are familiar with the machine inspection exit this is not there are some of says that you can also automatically generate this damp lets but is not always possible there is some work thus be done on that generating but normally if use time thighs damps let a pattern mat sufficient is occupied now another thing we started looking at when you say that either generate re-address code and from re-address code. I can convert that machine code get next optimization register and so on.

(Refer Slide Time: 08:22)



But I can also generate code by and we also at that or complete representation of some kind of where we note that exclusive okay , what you want to do this here is if you say I am trying basically issue what the technique is something like this that if I take countries .

(Refer Slide Time: 08:50)



Code Generation by Tree rewriting

- Target code is generated during a process in which input tree is reduced to a single node
- Each rewriting rule is of the form replacement → template { action}

  where
  - replacement is a single node
  - template is a tree
  - action is a code fragment

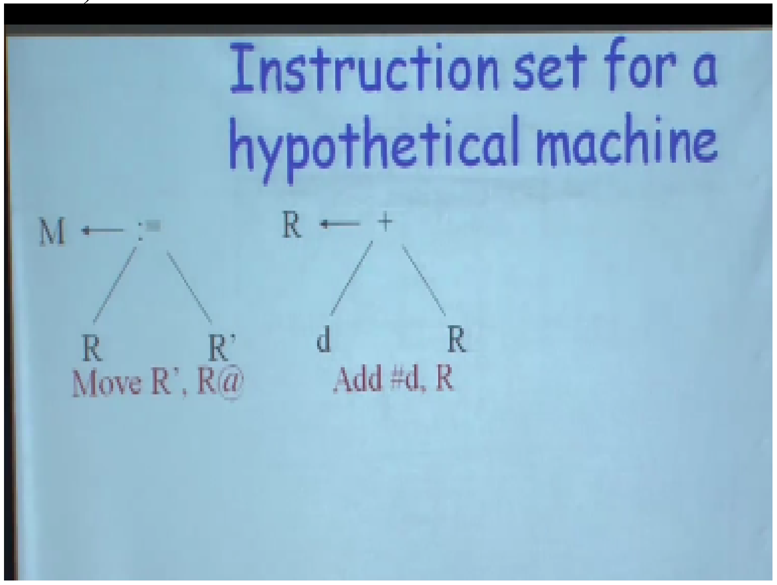
30

Suppose this is country I have and I will not go I will assume that all the subsides of this if in an say that I generate some code sequence for this suppose take something like this if I say that I am prime we to add or registered to registered suppose I have an instructions says okay this is a damp lets okay depending upon actual registers these could be any registered suppose I find that by template that in the trimly look something like this that may have a reduction that I have some value says excess value .

And have okay, so what I can do is I can find that this template actually something similar to this okay I know this instruction which will take value 1 and note what that I can do is I can have rewrite rule which says that if you find a template and know this template can be replace by something and the corresponding action as to be now this looks something very similar do what a country you had a damp let then you said that ,this damp let can be replace by left and side the damp let says going Eope.

Damp let replacing it and what was the corresponding action here ,corresponding action could I was another building extreme and I was code construction I was doing something so depending upon application I was unable to write some action so this looks very similar to that I have a template. Which I can replace something and the same time okay.

(Refer Slide Time: 10:48)



Instruction set for a hypothetical machine

Suppose now I have an instructions of this form which says is that I want to move on time register so what is template is saying that left hand side is an address right an hand side is register takes a value this assignment is being done and the target of this is oaky so this is a template which is saying that this instruction in general it says move contain of the register location tools addressing on the register okay.

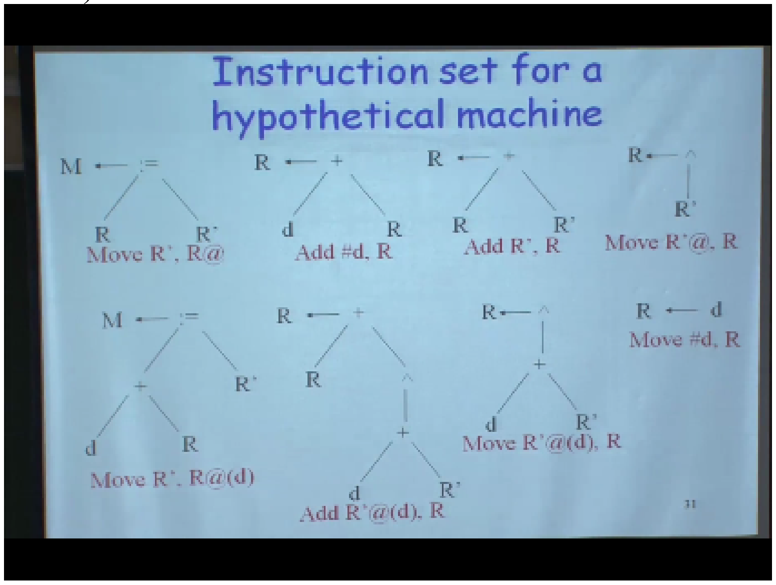
Deduction could be ,what are deduction could be that instruction find the damp let us and replace this sub-tree and something example of is done okay, but this is once possibility okay similar look at another instruction for example look this instruction says and content to register what this is saying is that I am looking at template is saying plus constant register and then I can replace the whole template by this register with by description of course.

I have to do actually which is so R is only but action part is will be the probably. Similarly I have for the instruction that activation are good valued for intend that hypothetical different. And what will be do is that refine the in which and depending has states okay. And keep of find this act will the single okay. So let us look at few more instructions one which we are done.

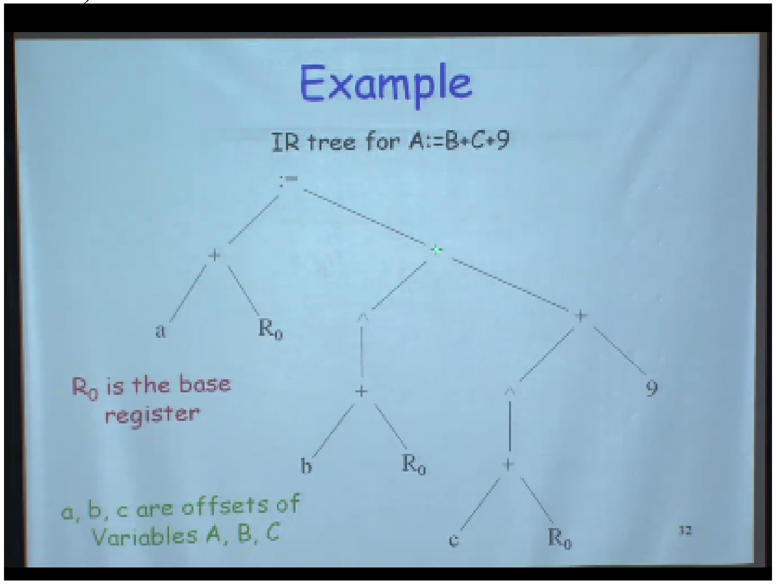
So here is various sometime we says contains of the various location to register and address of that the location will give the list of gun. So I can take these definitions to that contribute that code. So this basically is this is want a placement is and this is the actual generate the decision with instructions. Other possibility could be that I now say that who content of register location that is address is given by indirect which is that address will compute that various okay.

All I can say that I have done which says take content which locate that addresses given by add R prime with hypothetical B so this D and R prime that is me conduct of memory location and I am adding that R away. Yes would see we add memory location it could register and slow determine. And I could done of this episode move conducts of the memory location will take register so I can say that look at move contents of R prime at be do that at two declare oaky.

(Refer Slide Time: 14:53)



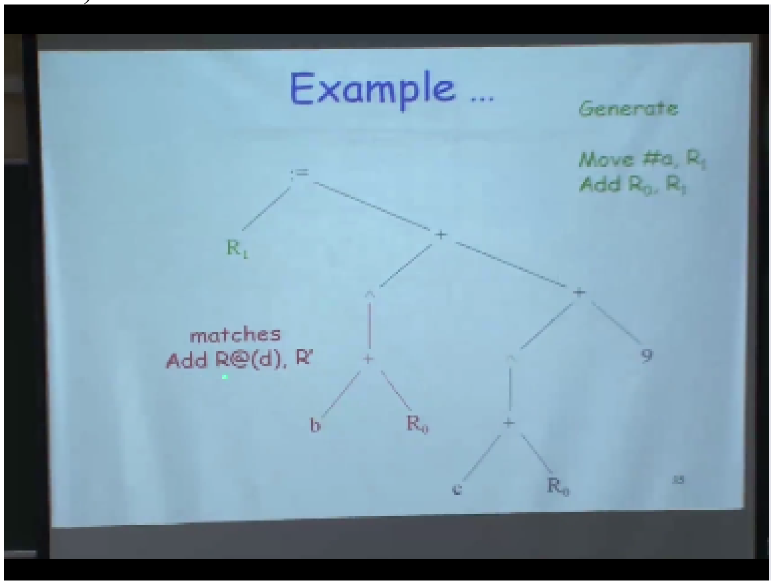
So this way I can decide template of each of this instruction so this get locate right we can point of at least I take machine instructions and right that the society.
(Refer Slide Time: 15:06)


Example
IR tree for A:=B+C+9
:=
+
a
R_0
+
^
+
b
R_0
+
^
+
c
R_0
9
R_0 is the base register
a, b, c are offsets of Variables A, B, C
12

And once have this okay now what can I do this I can take index tree and I can then spark with so first let us look at syntax tree for and expression like A=B+C+9 certain reception here one that all other in the sets whole. And therefore all the location which state book going to be same Ro is going to be A+B+C are the option of the store age of A, b, and c which is code to say I wanted this. Now what I need to this in this particular tree.

What is going this is saying that B+ Ro is giving by the wanted that location that value of B and the value of C had been add C+9 and I would slow this equal add this and address to the left hand side which is R0+ A and Ro. So what are the kinds of do this okay now I can match several compacts and the calculate are use are follow these samples. Now remaining the machine you have can be different what we can do is start from here.
And first we says that look at this scale and this scale I doing totally order okay but the purchase look at this role this says that matches code says part the first for use is R1 because R0 is already block then outside. And now generate the instruction is Ro and R1 and this code is this E is replaced by now this constant will be rule this instruction and this is how proper. And now again I try to do and say that now this part of the tree matches add traditional and R prime and R0 is generate the instructions.
So now I can said that generate for instruction say Ro can be because Ro can be uses here that is can be used here. So what I am doing here is and R1 and then what will do I am going to replace

this particular note by this case okay. Now I can find mode here and there is one possible calculate which say that at I am adding.
(Refer Slide Time: 18:29)



Two locations are I am writing this particular location is R1 so what my do now I just say that can the distraction says that actually move instruction saying that I moving content of this include R2 and this is look I will correct it okay. This is induct and then what we can do is we can go for the and we can say that this is again the same. So again is doing here is that you are moving of this okay.
And so I am looking at indirect R0 and C I am adding the two that adding C there is nothing like R1 and R2 what would be initialize to that will move instructions complete so this becomes R3 this is after four conditions okay and then what will happening that I will now match another calculate is now add the constant to arranges code okay. So this constant to be hearing distract this is the calculate.
What will happened there is that I can now generate this instruction with says that add B at R3 and I once do that then the spot will and once have done this okay what will happen now it will match may be tentacle which is act to instructs and store value in one other basically this not happing and matching R3 and that will generate something like add we already added R2 in once have this okay.
This is now going to match calculating say that add R prime R to be indirect and this is move R to In this should be add root okay. And I am saying that actually this template this is incorrect but this move is correct equal adding is moving I just anything in this R2 in indirect this instruction

okay. And this is by instruction sequences I generate what why replaced it with it replace on the topic of this instruction is whole thing is memory and I can stop.

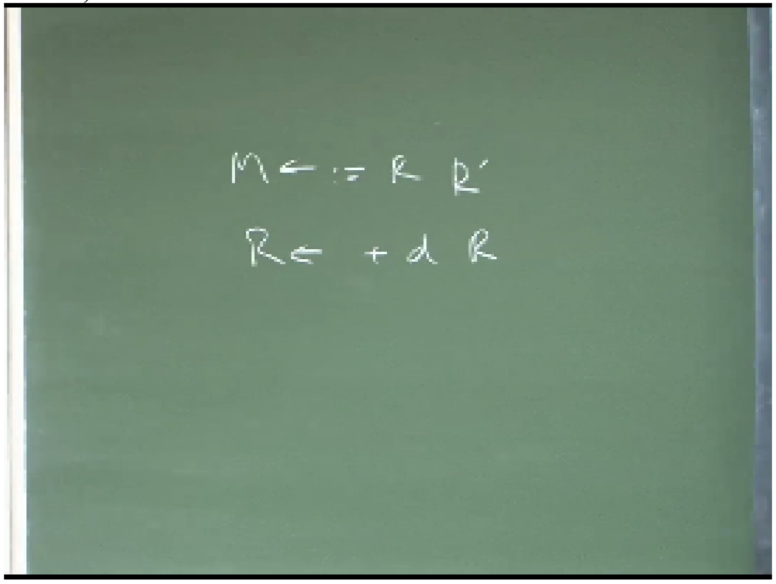
But a several issue of what is the implement of this because we are do that in the second issue is given more that is I adding this. Let us look at sequences now could have return look at the sequence have any list of lift hand side okay. Operator is communicate that means I can also I contest will say that left hand side in the discoursing in the right hand side. So if I know go back to the completes are add so let us go back to the clear templates we actually wrote okay.

What we actually wrote for something like this okay that I am adding the content in the pattern beyond with communicate the either D lane for right okay. I do not know the user code is going to me and user sign +B and move I should be generate and efficient code and on the both instructions does not matter nature then when have a like this and I calculate which nagged +RD are have a way of saying this is operator with communicated four demonical right.

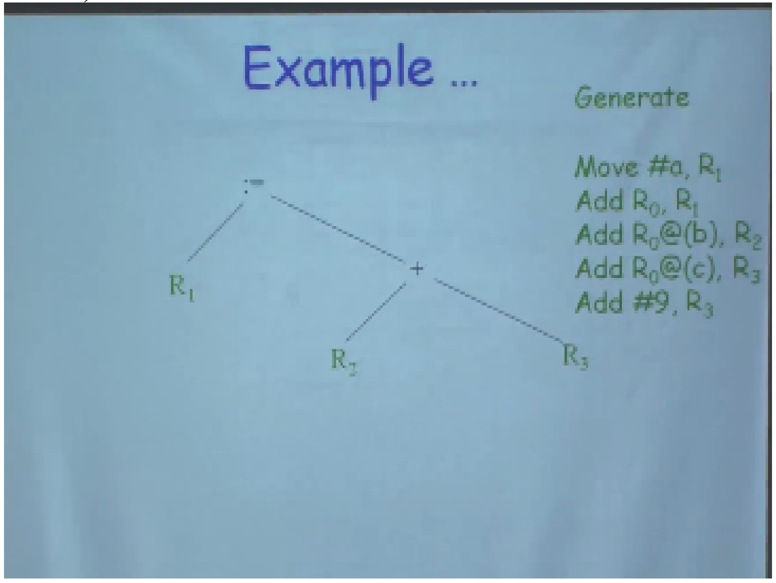
And now if I look at the these become first because now the number of compensations you can generate form this particular instruction of so I can take I can slap on this okay move look at these instruction again generate particular axis. But at the ways of ending by how do I match look the cementation in and you are instruction that equation to be how for implement that was of some tree packing of variables.

You know something that t matching and the t so far none what is gate the gate be that to match suppose I do now a carves allow this pattern set the re orders diverse re order caveperson and just plat in it write with the one of the form it says free after carvers and so can I say if something like this and is define.

(Refer Slide Time: 24:44)



Can I flat it like this okay all can I take second instructions and say r is define okay and this know but becomes like context free grammar and then whatever free I have for AB will assign A plus B plus c plus remain and that can also be there was that b out of the carves and flat end and then what can I do can I use when two just merging is that merging okay so keep merging it not be only in this way you can use even two type and to actually do code generation okay
Once have generated into the representation I can write my machine instructions and now like grammar context to the grammar and then can do a pattern just using to add code from this and what is the action we have write here we already know the action and we write to here is some book keeping for register and then generate some instruction and okay.
I can write this kind of action by the single table to do and the situation okay only problem is going to once again that different there was a therefore different way of flattening this particular tree are going to give me different code sequences for example the code sequence.
(Refer Slide Time: 26:17)



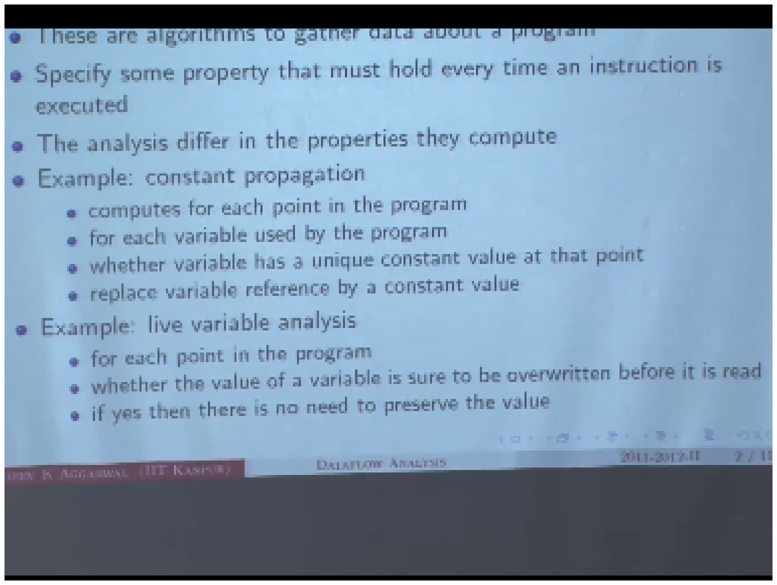
Example ...

Generate

Move #a, R_1
Add R_0, R_1
Add R_0@(b), R_2
Add R_0@(c), R_3
Add #9, R_3

We look were form this maintain the representation which was this code sequence with these two errors instead of add this should have been move okay I can take this same machine instruction and then generate different code sequence for that if I start doing my matching from the right hand side so suppose now I start matching my template from the right hand side and I say that this matches template which is move constant to register and then I generate this instructions in says to right to a register and the next pattern could be something like this okay.
Were actually match this whole thing and this is saying that now and this value into register r prime and r prime is now in this case r1 and then I can generate this instructions which says r0 I n direct c to r1 okay that gives me c plus 9 okay and that I replace the whole thing pattern r1 and

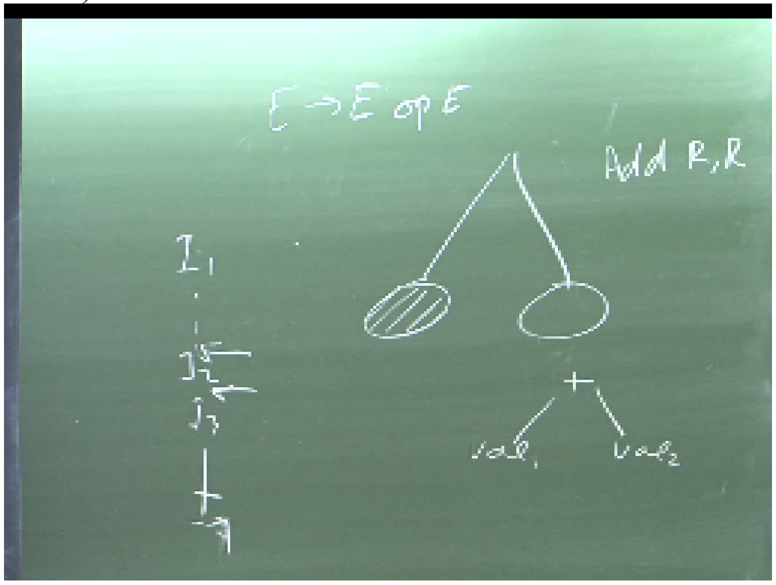
then the same pattern actually matches here and what I can do here is I can say that now this particular template matches this instructions template and what I can generate is add r o d r1 and this whole thing can replace by r1 and this whole thing.
Can replace with r1 template which is saying that move r into a location who address is given by r1d and here d is contently what are thing generate is move r 1 going to r0a directly so users required so instead of those instructions are generated I can just generate four instructions okay. So cleversal order this going to matter a lot and what immediately why has to think about this for looking issue of cleversal so that lot of techniques people have done work on this saying that irrespective of whether you operator this consultative or not okay .
And how do you do fact and manage and how you really do something about reducing the cleversal order and generate at least ensure that will generate construction sequence okay which is that okay so there lot of results are available on this people have used different technique in fact this is one of the pattern matching remain the most power full techniques as far as different code generate techniques okay.
Many compilers actually instead of three address code to do this kind of template matching and generate code and you can see that this technique also is very enable to retargatable code generation because all need to do this right this template are different machine and once we have this templates or a different machine when I can take same intermediate presentation I can take the same code generation algorithm.
And that I can there was over this particular intermediate presentation and then generate code by bring a pattern match with respect to a new machine template okay so you can do so this is a way I want to stop one code generation okay and I want to move on to a new topic which obviously we will know it able to cover in detail just have a 20 minutes but I will give you glance atleast into what happen in optimization.
The optimization is something which is clearly formal stuff okay let us look at something atleast I am giving an ideas of what optimize is all about okay so we talked about this problem called data flow analysis right I mean I mention this term okay.
(Refer Slide Time: 29:50)



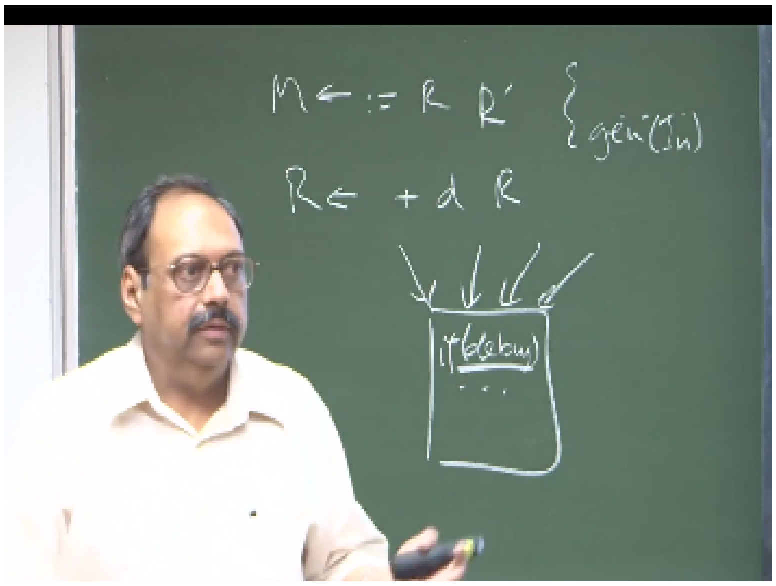
- These are algorithms to gather data about a program
- Specify some property that must hold every time an instruction is executed
- The analysis differ in the properties they compute
- Example: constant propagation
  - computes for each point in the program
  - for each variable used by the program
  - whether variable has a unique constant value at that point
  - replace variable reference by a constant value
- Example: live variable analysis
  - for each point in the program
  - whether the value of a variable is sure to be overwritten before it is read
  - if yes then there is no need to preserve the value

I mention this term data flow analysis okay and then we will say that into optimization okay so what is data flow analysis first let us look at once actually and then I can flow some without symbols and something toward so what we have to time to do this we are looking at the program and we are trying to find weather certain properties hold for this program what is my input data and what is the passed which have been taken during execution phase of this particular program.
For example if you recall okay in the previous once we had this d was had define to 0 and I had this kind of hash define d goes to you okay and then I had this debug are does not matter past is state if with this variable which is define only once then all places irrespective of your input at all the places became replace by the 0 okay how do I know this okay I have to do certain analysis over the program to say that.
At the statement which says if d - and go to add 2 there is the multiple parts which has reaching here that does not matter which part in fallow during execution the value of d- will reach here is all this going to be 0 if I can assign that a compile time there is property is independent of what happens are execution time so what we will doing data flow analysis is we try to find out certain property of the program during compiler.
And see that weather these properties can be use from of program analysis that are the data flow analysis okay so global optimizations and remember that only optimization is done for this looking at the local basic block and using next use information’s and d saying that I can minimize my number of resources for looking at this information.
Suppose I want to go beyond this sequence okay then I need to worry about multiple path which means is basic logics okay so basically global optimization are all based on we are all known as data flow algorithms and what these algorithm do the gather certain data about that program and

then what is dad is going to tell us that we are looking for certain property that was fold every time this Instruction.
So the property was always the property always by 0 this must always irrespective of the entire program that a 0 okay and that was the data flow analysis able to tell me and for different property I want to compute the analysis may be far okay.
Let us look at one situation where I say that I want to do constant propagation so constant propagation I said is define is initialize to 0 and then I was saying is debug and so on okay.
So what we are all do for the constant propagation we want to find out for every point in my program now what is point in a program point in a program is nothing but one of the basic blocks we constructed and instructions which is getting into it okay so her the number control flow that I constructed of the program by saying that I can find that then I convert the break that tree structure sequence and to a set of basic laws and then have a control flow okay.
So during a execution I will need either before that instructions or either after some instructions these are high program point if you recall when I am was doing this next to instructions computations.
(Refer Slide Time: 33:11)



I had this instructions in I2 , I3 and so on okay and I was doing computation of this property saying that at this point of time and this point of time weather to variable is live or dead okay so this is the information I w as computing so now we say that each point of time and program and for each variable which is the use in this program weather by variable has a unique content value and point and if this condition is true if it is property is true then I can replace variable reference for context variable.
(Refer Slide Time: 33:52)

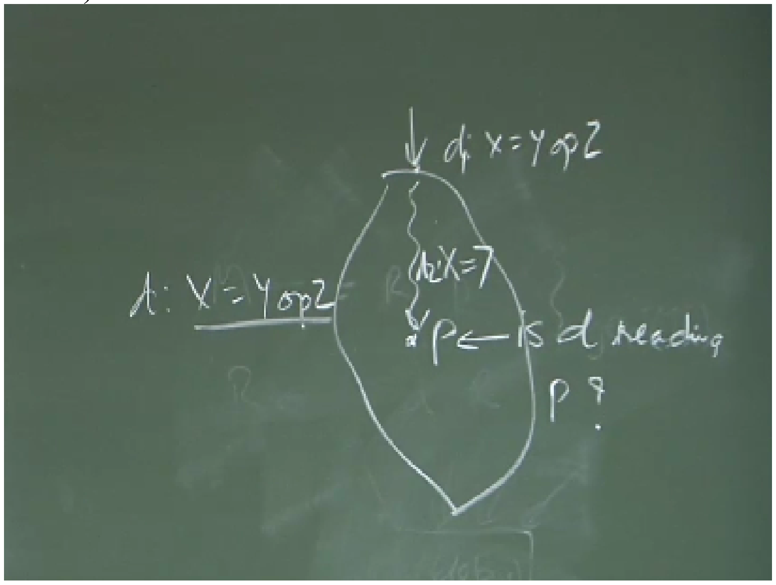


For example we will say that if I can look at the basic law which contains the instructions which is if d bug and do something and there are multiple path which are coming onto the basic law in my control graph if I can assert by analysis of this program that does not matter which path is taken irrespective that this program point value of d bug is always be 0 okay.
Always say does not matter which the path is taken that means I must check this property in all path before I can do this secession okay if I can check that and this debug always turn at to be 0 then I can always replace his value by 0 right so this is what we do normally in constant configuration and check this property at every program point saying that look at all possible paths and find out what is the value of debug.
And if it is a unique value which is the same okay I can always do this replace okay or another thing is we saw was this line variable this line variable now line variable we are looking only within a basic block and suppose we have a control flow that means I was able to check that if I go out and multiple path from this okay.
Weather this variable debug some variable which is available here to use again what I need to do here is that for very program point I want to find the weather value of variable is sure to be overwritten before this is going to be read again okay and I yes then there is no need to okay and if yes then there is no need to preserve that value okay if I say that if I have a variable x here and if I know that does not matter which path I take then this value is always going to be over written one of this paths and all of this paths before next this happen okay then I know that whatever value as been a access flow is no use okay.
And I can that just in move it okay so what I am doing in next to information’s was assuming that we using controlling flow path and that was the basic property of basic blocks and the single

entry but can I go into data flow analysis I say there are the multiply entry and the multiple exits and this property must allow the check for all possible paths now you can see that immediately one problem that comes it talk about the reason is could be infinite and also if you loop for examples if I say that then the note.
Where I end of the control this note where I exist the control mad this is going to be unique to be program then I can count number of parts in that and this part may be number of phrase may be finite I have atop infinite form suppose I cannot effort anything about whether you terminate or not now exactly check this all possible parts or I will say I can do some clapping and say that if I check this property for some of the parts then I can be sure about the property all the parts right because somewhere you have to say that whatever form of your solution is it is in close form I just cannot finite and parts check in this property.
Possible parts that is not available and even if I large number of parts again computation so costly and not given and so I have do this kind of analysis since I am just giving you what happens here let me skip some of the property material and straight away come to definition which will actually tell you something about that how to module analysis, saying that although I have infinite part.

But I can talk only about finite property okay, so what we say is that we define something called region definition so what do here is suppose I have upper control photo graph and this control program

(Refer Slide Time: 37:44)



This is program .T and this is some entry into control okay now on to find out something like chain that this is a definition D so what is my definition if you recall again definition saying that if in general I have instruction like this which says x is be assign y (z) this is given a name makes this instruction I or we definitions of X we say that definition we region this is a point D if it is a pats D to 3 and these not held on this form.

Let me this values is not over it suppose on this path I have this program point definition D which is defining at X and here I want to say if D reaching P okay, how do I do that this D will be p now suppose which is only one path okay, wrong to p and on this path I overwrite I say X is the sign what I have let us say this is D 1 and this is D2 and I say D1 ever reach p okay D1 will never reach p because on this path it is always going to be overwrite so this is one set of information.

When I am trying to do deduct to analysis whether I have a path from any definition to point in the program p and the formal map doing that is just not you wrote program and it start going this correctly similarly I may say that what to expression which are available to suppose I want to do form of the expression okay, suppose I am saying that at this point so now let me change my formalism A is assign X+Y okay.

And suppose they are multiple path coming here can I now look for property which will, say that is X+Y always computed on all these paths because if I have say a computation of X+Y on this path than I known for shown that does not matter which path comes here X+Y has been computed now it may not have same value okay and this path it may have a different value but it that bother.

As long as we know that X+Y has been computed all the parts okay, then I know that compute X+Y again so I can eliminate this second computation of x+y because I know that respective of which path is taken during execution I will have some value of x+y so again I can do this kind of replacement and we say that the expression x+y is available at point p if every path p evaluate and I do not make an assign and imagine that in am computing x+y on this path and immediately after this I am saying that as soon as I make an assignment to x this x+y becomes invalid .

Because the value I will be computing here will not be this value but will give something else okay so again looking another property may be I want to check whether the variable is live at a point p and we can say that the value of x can be used along some path starting from p so what may happen is that suppose I do this computation okay I say that I want to reuse this computation and suppose this path is taken then I cannot use this value so this value has been already been changed while assign to x .

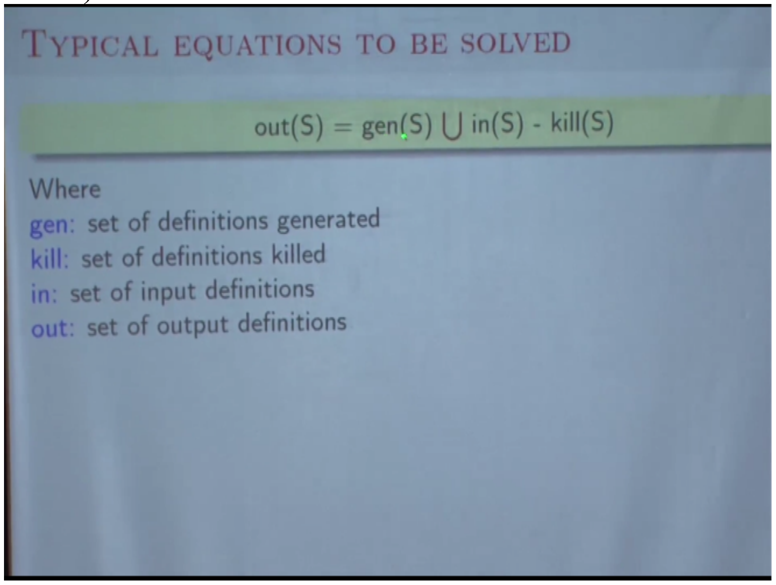
But suppose this path was taken okay then I can use this value but I do not know at compile time which path will be there this path could be taken so I want to assert this property over all these paths okay so I should not have any assignment to either x or y and if you have any assignment then you need to have one more computation so I can say that if I have my control flow graph okay and I want to say x is live here at this point p then what will I say .

If it can be used some path so suppose I have multiple paths going out here okay and if it I used on one of the paths then I know that surely know that it is a graph now you can see something interesting in these three definitions the first definition if you see this says that if something happens on one of the paths okay because there are multiple paths here what is happening here you say so now it is possible on the other paths I may not even have a definition of x but at least so I do not know which path is going to be taken here okay this path could be taken probably then I know at least as far as this point is concerned this is an reaching definition.

But in this case I am assigning that something has to happen all the paths that if x+y is available

here then it must be computed on all the paths because if it is computed on only one of the paths okay and the control flows to this path at run time and I know that x+y will not be available so it must be computed on all the paths for this to be reaching definition this is sufficient to say that this available on one of the paths .
Similarly when I say that this variable is here and I do not care about the paths if I say that is variable I live even if one of the paths uses it then I know that variable is live so this is now what I am computing here is that suppose I say x is used on this path and I assert that x is live here but actually at run time this path is taken then I know that actually this value will be used but I am looking for certain properties so I am trying to compute and so I can do run time optimization .
So what I am trying to do first is the compile time and I know for certain properties which is always good okay is this clear what are they doing here so I may write something interesting I may say that each basic block can generate a set of instructions and each basic block.

(Refer Slide Time: 45:01)

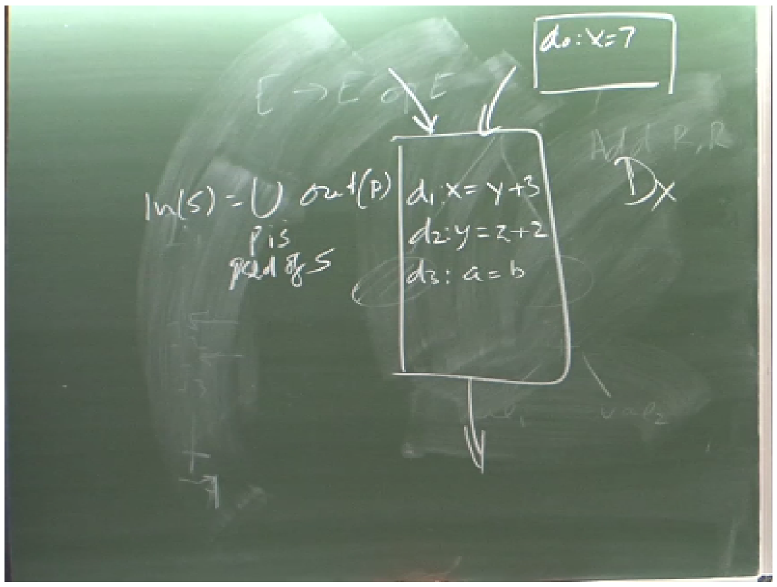


Typical equations to be solved

$$out(S) = gen(S) \bigcup in(S) - kill(S)$$

Where
gen: set of definitions generated
kill: set of definitions killed
in: set of input definitions
out: set of output definitions

(Refer Slide Time: 45:23)



So suppose I write a basic block in which I have set like d1 in which say x is assigned y+3 and instruction d2 we say that y is assigned now z+2 and I have d3 which I say a=b and suppose I say that we have these instructions now I know that when control comes in here so what I know for sure is that irrespective of which paths are taken and the end of these contains only these basic block instructions and in the end of this I have a definition of x and this is definitely this basic block is going to generate three definitions.
So for example I have a basic block and I have a value x assigned to do and even if there is a path from here to here that if the control flows there then this do will not be available so I can compute now set of definitions which are generated by basic block I can compute certain definitions which are always going to be used by basic block okay so for example I can say that if my universal set of definitions of x is Dx then what are the definitions which are going to be used by basic block .
So it is going to kill all the definition which are dx so no matter how the control flows if any definition comes so I can compute my sets of variables which are generated and the definitions which are generated and then I can write equations which are saying that the definitions which are coming in what are the definitions which will come out so the definitions which will come out are either it is generated in the block or it comes into the block .
Okay I can write certain definitions for each basic block and example for each definitions we are trying to say then I can write an equation which will be of thee form that what comes out of a basic block is nothing but what gets generated or what comes in so what we will try to do is we try to assert these kind of properties over a program and then use these properties and this is just an glimpse to give you what happens to optimization.

The optimization is a very formal structure and not pattern match saying that something will not work and you set up these kind of equations now you can see that if I have one more equation could be that what is my inset if I compute my inset o inset in reaching definition could be that I take union out of all the basic blocks where p is set of s we finally say that if I take a union whatever is generated and this way for each basic block and so let us close our discussion here today.

Acknowledgment
Ministry of Human Resources & Development
Prof. Phalguni Gupta
Co-ordinator, NPTEL IIT Kanpur
Satyaki Roy
Co Co-ordinator, NPTEL IIT Kanpur
Camera
Ram Chandra
Dilip Tripathi
Padam Shukla
Manoj Shrivastava
Sanjay Mishtra
Editing
Ashish Singh
Badal Pradhan
Tapobrata Das
Shuubham Rawat
Shikha Gupta
Pradeep Kumar
K.K Mishra
Jai Singh
Sweety Kanaujia
Aradhana Singh
Sweta
Preeti Sachan
Ashutosh Gairola
Dilip Katiyar
Ashutosh Kumar
Light& Sound
Sharwan
Hari Ram
Production Crew
Bhadra Rao
Puneet Kumar Bajpai
Priyanka Singh
Office
Lalty Dutta
Ajay Kanaujia

Shivendra Kumar Tiwari
Saurabh Shukla

Direction

Sanjay Pal

Production Manager

Bharat Lal

an IIT Kanpur Production