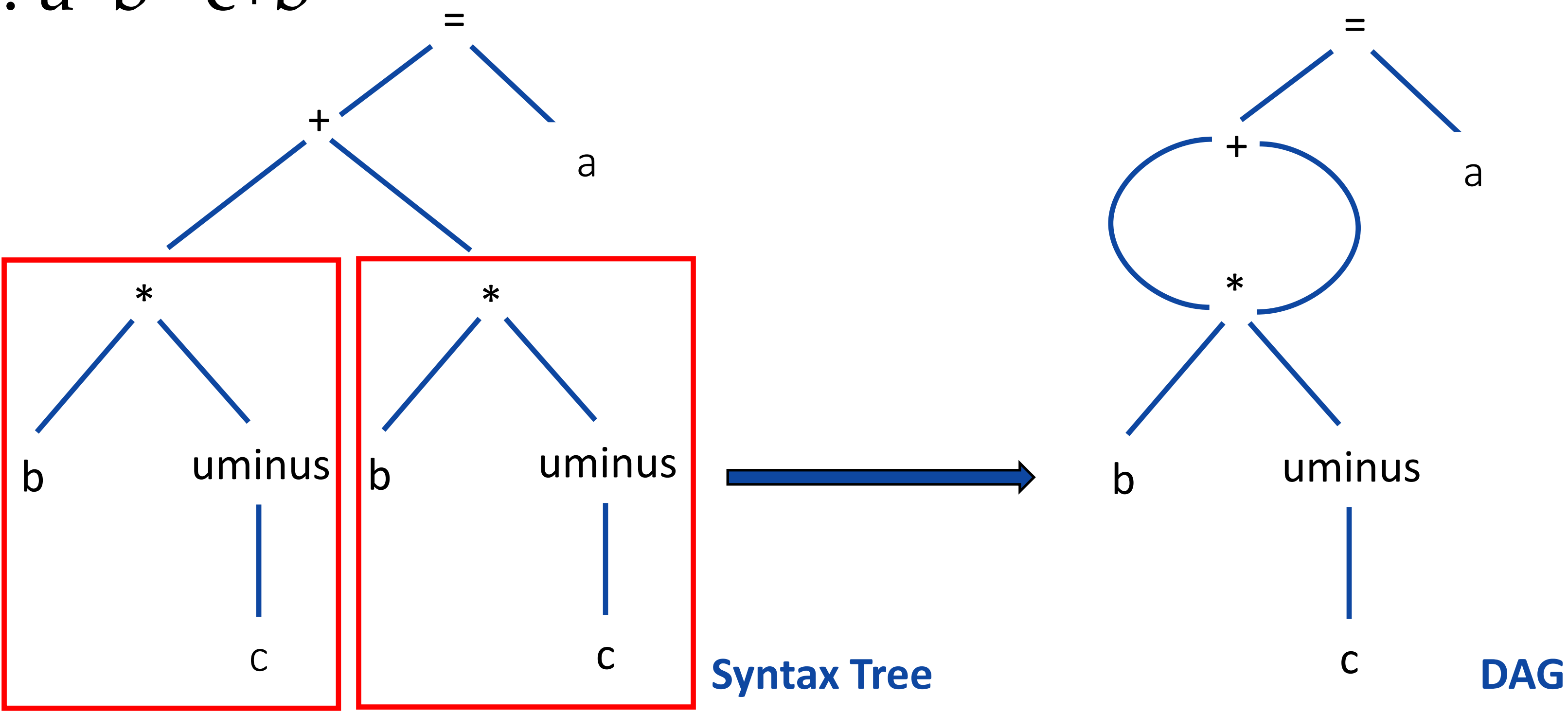
Module 4 – Intermediate Code Generation

Different Intermediate Forms

- Abstract syntax tree
- Postfix notation
- Three address code

Abstract syntax tree & DAG

- A syntax tree depicts the natural hierarchical structure of a source program.
- A DAG (Directed Acyclic Graph) gives the same information but in a more compact way because common sub-expressions are identified.
- Ex: a=b*-c+b*



Postfix Notation

- Postfix notation is a linearization of a syntax tree.
- In postfix notation the operands occurs first and then operators are arranged.
- Ex: **(A + B) * (C + D)**

  Postfix notation: A B + C D + *

- Ex: **(A + B) * C**

  Postfix notation: A B + C *

- Ex: **(A * B) + (C * D)**

  Postfix notation: A B * C D * +

Three address code

- Three address code is a sequence of statements of the general form,

  a:= b op c

- Where a, b or c are the operands that can be names or constants and op stands for any operator.
- Example: a = b + c + d

  t_1=b+c

  t_2=t_1+d

  a= t_2

- Here t_1 and t_2 are the temporary names generated by the compiler.
- There are at most three addresses allowed (two for operands and one for result). Hence, this representation is called three-address code.

Different Representation of Three Address Code

- There are three types of representation used for three address code:
  1. Quadruples
  2. Triples
  3. Indirect triples
- Ex: $x= -a*b + -a*b$

  $t_1 = - a$

  $t_2 = t_1 * b$

  $t_3 = - a$

  $t_4 = t_3 * b$

  $t_5 = t_2 + t_4$

  $x = t_5$

Quadruple

- The quadruple is a structure with at the most four fields such as op, arg1, arg2 and result.
- The op field is used to represent the internal code for operator.
- The arg1 and arg2 represent the two operands.
- And result field is used to store the result of an expression.

$x = -a*b + -a*b$

$t_1 = -a$

$t_2 = t_1 * b$

$t_3 = -a$

$t_4 = t_3 * b$

$t_5 = t_2 + t_4$

$x = t_5$

Quadruple

No.	Operator	Arg1	Arg2	Result
(0)				
(1)				
(2)				
(3)				
(4)				
(5)				

Triple

- To avoid entering temporary names into the symbol table, we might refer a temporary value by the position of the statement that computes it.
- If we do so, three address statements can be represented by records with only three fields: op, arg1 and arg2.

Quadruple

No.	Operator	Arg1	Arg2	Result
(0)	uminus	a		t_1
(1)	*	t_1	b	t_2
(2)	uminus	a		t_3
(3)	*	t_3	b	t_4
(4)	+	t_2	t_4	t_5
(5)	=	t_5		x

Triple

No.	Operator	Arg1	Arg2
(0)			
(1)			
(2)			
(3)			
(4)			
(5)			

Indirect Triple

- In the indirect triple representation the listing of triples has been done. And listing pointers are used instead of using statement.
- This implementation is called indirect triples.

Triple

No.	Operator	Arg1	Arg2
(0)	uminus	a	
(1)	*	(0)	b
(2)	uminus	a	
(3)	*	(2)	b
(4)	+	(1)	(3)
(5)	=	x	(4)

Indirect Triple

	Statement
(0)	(14)
(1)	(15)
(2)	(16)
(3)	(17)
(4)	(18)
(5)	(19)

No.	Operator	Arg1	Arg2
(0)	uminus	a	
(1)	*		b
(2)	uminus	a	
(3)	*		b
(4)	+		
(5)	=	x	

Exercise

Write quadruple, triple and indirect triple for following:

1. -(a*b)+(c+d)
2. a*-(b+c)
3. x=(a+b*c)^(d*e)+f*g^h
4. z=g+a*(b-c)+(x-y)*d