HIT Harare Institute of Technology

success through innovation

H180072B

Pupurayi Paula Marisa

Compiler Design

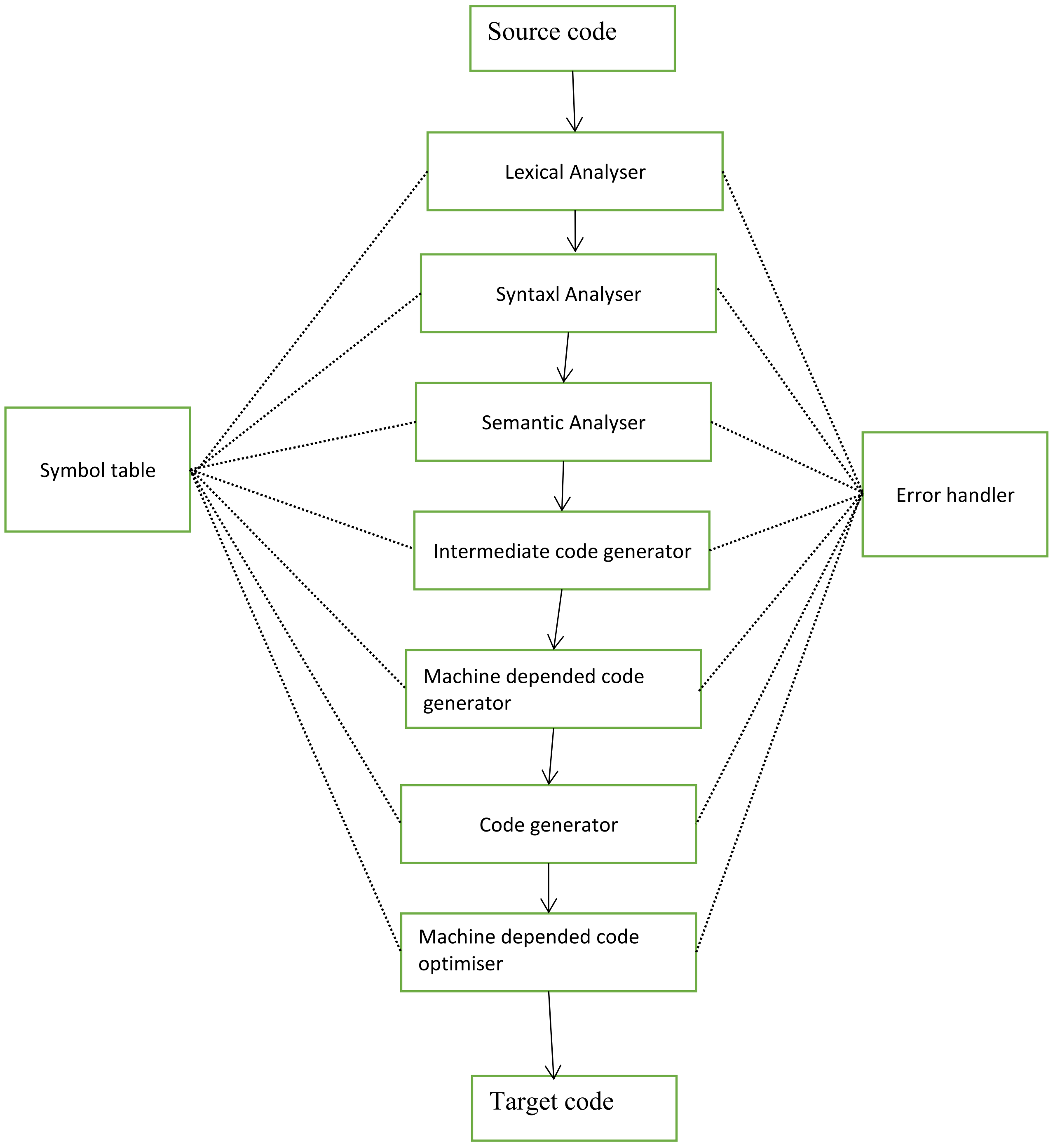
Software Engineering Department

Test 1

a) Write a Regula Expression for a language containing the strings which end with "abb", over $\sum$ ={a,b}.

RE = (a+b)*abb
RE= ((a+b)^+)b

b) i What are the various phases of a compiler.



ii Explain each phase in detail.

Lexical analyzer
It works as a text scanner, scanning the source code as a stream of characters and converting it into meaningful lexemes. Lexical analyzer represents these lexemes in the form of tokens as: <token-name, attribute-value>

Syntax analyzer
This phase is called the syntax analyzer or parser. It takes the token produced by lexical analysis as input and generates a parse tree (or syntax tree) as the output. The token arrangements are checked against the source code grammar, i.e., the parser checks if the expression made by the tokens is syntactically correct and acceptable.

Semantic analyzer
Semantic analysis is checking whether the parse tree constructed (by the syntax analyzer) follows the rules of language. For example, assignment of values is between compatible data types, and adding string to an integer. The semantic analyzer also keeps track of identifiers, their types and expressions; whether they are declared before use or not, etc. producing an annotated syntax tree as an output

Intermediate code generator
After semantic analysis, the compiler generates an intermediate code of the source code for the target machine. It represents a program for some abstract machine. It is in between the high-level language and the machine language. This intermediate code should be generated in such a way that it makes it easier to be translated into the target machine code.

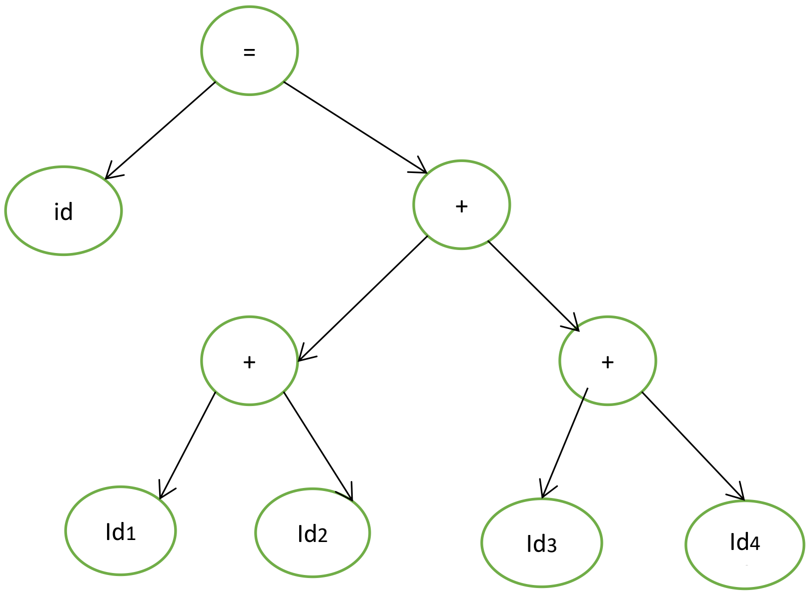
Code Optimization
The intermediate code is the one which is optimised. Optimization can be assumed as the removal of unnecessary code lines, and arrangement of the sequence of statements in order to speed up the program execution without wasting resources (CPU, memory).

Code Generation
In this phase, the code generator takes the optimized representation of the intermediate code and maps it to the target machine language. The code generator translates the intermediate code into a sequence of (generally) re-locatable machine code. Sequence of instructions of machine code performs the task as the intermediate code would do.

Symbol Table
It is a data-structure maintained throughout all the phases of a compiler. All the identifiers' names along with their types are stored here. The symbol table makes it easier for the compiler to quickly search the identifier record and retrieve it. The symbol table is also used for scope management.

biii)Write down the output of each phase for the expression : x=(a+b)+(c+d)
Token stream
$Id=id_1+id_2+id_3+id_4$

Syntax tree



Semantic tree

The result of the semantic tree is the same as that of the sytax tree. This is because the syntax tree already follows the rules of the programming language and has the identifiers in the right texts.

Intermediate code

$t_1=id_1+Id_2$

$t_2=id_3+Id_4$

$t_3=t_1+t_2$

$Id=t_3$

Optimised code

$t_1=id_1+Id_2$

$t_2=id_3+Id_4$

$id=t_1+t_2$

Generated code

```
MOV id1,R2
MOV id2,R3
ADD R2,R3
MOV id3,R4
MOV id4,R5
ADD R4,R5
ADD R3,R5
MOV R5,id
```