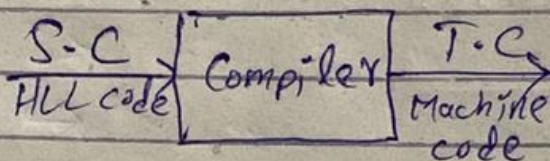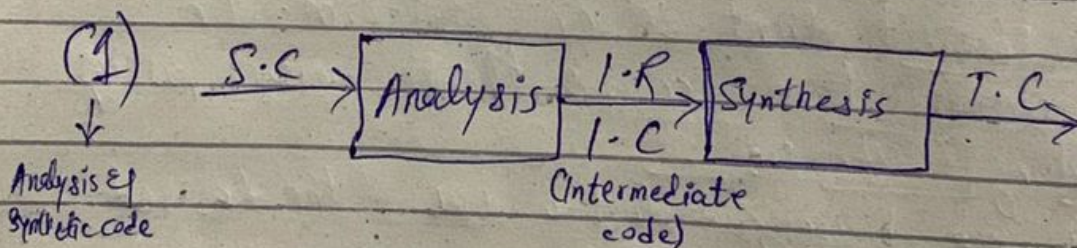## Compiler

(Def)
Compiler: Compiler is a system software that takes some source code as input, analyze it and translate/convert it into some target code (machine code)

$$\underset{\text{HLL code}}{\underline{S \cdot C}} \rightarrow \boxed{\text{Compiler}} \xrightarrow[\substack{\text{Machine} \\ \text{code}}]{T \cdot C}$$

Decomposition of compiler:

1) Analysis-Synthesis Model (diagram is incomplete in notes)
2) Logical Phases
3) Front-end & back-end
4) Pass

(1) $\underset{\downarrow}{} \xrightarrow{S \cdot C} \boxed{\text{Analysis}} \xrightarrow[\substack{I \cdot R \\ I \cdot C}]{} \boxed{\text{Synthesis}} \xrightarrow{T \cdot C}$

Analysis &
Synthetic code

(Intermediate code)

→ **Lexical Analysis:** only done if reserved words, punctuation marks are all checked. Words are checked against lexican (vocabulary) of language.

→ **Syntax Analysis:** checking the syntax.
(sense/meaning)
→ **Semantic Analysis:** Checking that the ~~syntax~~
~~used by~~ code ya have written, does it make sense or not.
Hamid is reading the book (✓)
Book is reading the Hamid (✗)

$s/3 (\checkmark)$ but $s/o (X)$

$x = y + z$ is semantically correct
but $z + y = x$ is semantically incorrect as calculation
is supposed to be done on RHS, not on LHS.
In analysis-synthesis phase,
↑ The compiler is decomposed into two phases,
analysis phase & synthesis phase.

1) Analysis Phase: The analysis phase takes
the S.C as input and perform lexical
analysis, syntax analysis and semantic analysis
and convert the S.C into I.C (intermediate
code).

2) Synthesis Phase: The synthesis phase takes the I.C
as input and translate it into machine code/
target code (T.C).