CS306
PPL CSE
Tech 6th Sem

Principles of Programming Language

__/__/____

Subhojit Ghimire
1912160

* System Language : 1960 → CPL (combined Prog. Lang.)
  → 1966 → BCPL (Basic CPL)
  → 1972 → C

* Level of PL:
  1. Machine Lang. : Binary → 0, 1
  2. Assembly Lang. : MOVE, ADD SUB
  3. High level Lang. : C, C++, Java
  4. 4GL Lang. (4th Gen Lang.) : VC++, SQL, Oracle

* Why study Programming Language:
  → To improve ability to develop effective algorithm.
  → To improve use of existing PL.
  → To increase vocabulary of useful Programming Constructs
  → To allow a better choice of PL.
    → Numerical Application/Calculation → FORTRAN
    → AI Application → LISP
    → Internet Application → JAVA
    → System Programming → C
  → To make it easier to learn new languages.
  → To make it easier to Design new languages.

* Programming Language Paradigms:
  → Imperative or Procedural Languages : C, FORTRAN
  → Applicative or Functional Languages : LISP;
  → Rule-Based or Logical Languages : CLIPS; JESS
  → Object-oriented Languages : C++, Java, Python
    → Encapsulation, Data Abstraction, Inheritance, Polymorphism, Message Passing, Extensibility

__/__/____

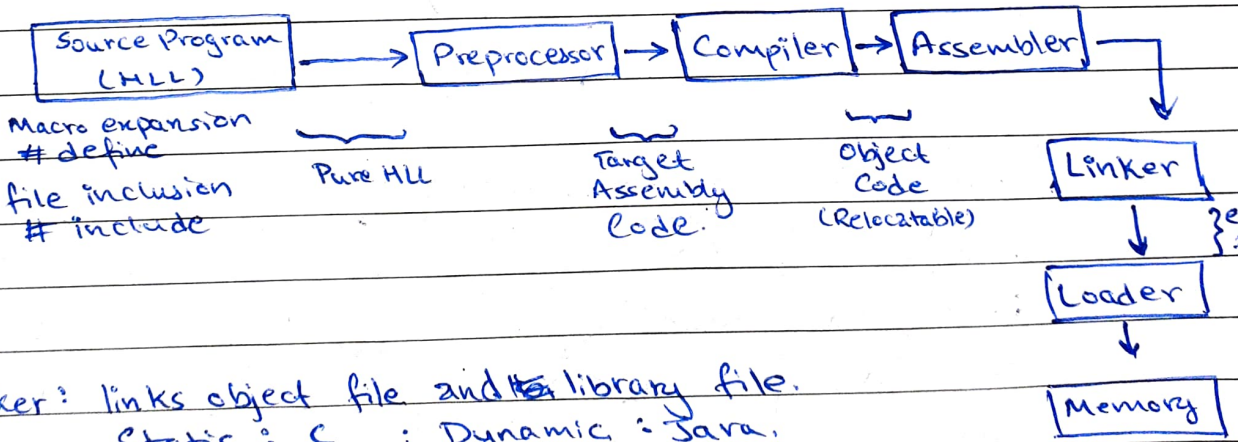
✱ Language Description (Syntax and Semantics)

- Syntax specifies how programs are built-up.
- Semantic specifies what program means.
- Expression Notation : Prefix, infix, postfix, minfix notations.
- Abstract Syntax Trees
- Lexical Syntax
- Context Free Grammars : BNF; Syntatic Ambiguity; Dangling Else Ambiguity
- Grammars for Expressions :

✱ Language Translation Issue :

input (HLL) → [Translation and Execution] → Output

- Programming Language Syntax : Character Set, Identifier, Operator, Symbol, Keyword, Comment, Delimiters, Expressions, Whitespaces, Syntatic Criteria.
- Stages in Translation (continue on next page).



Linker : links object file and library file.
Static : C ; Dynamic : Java.

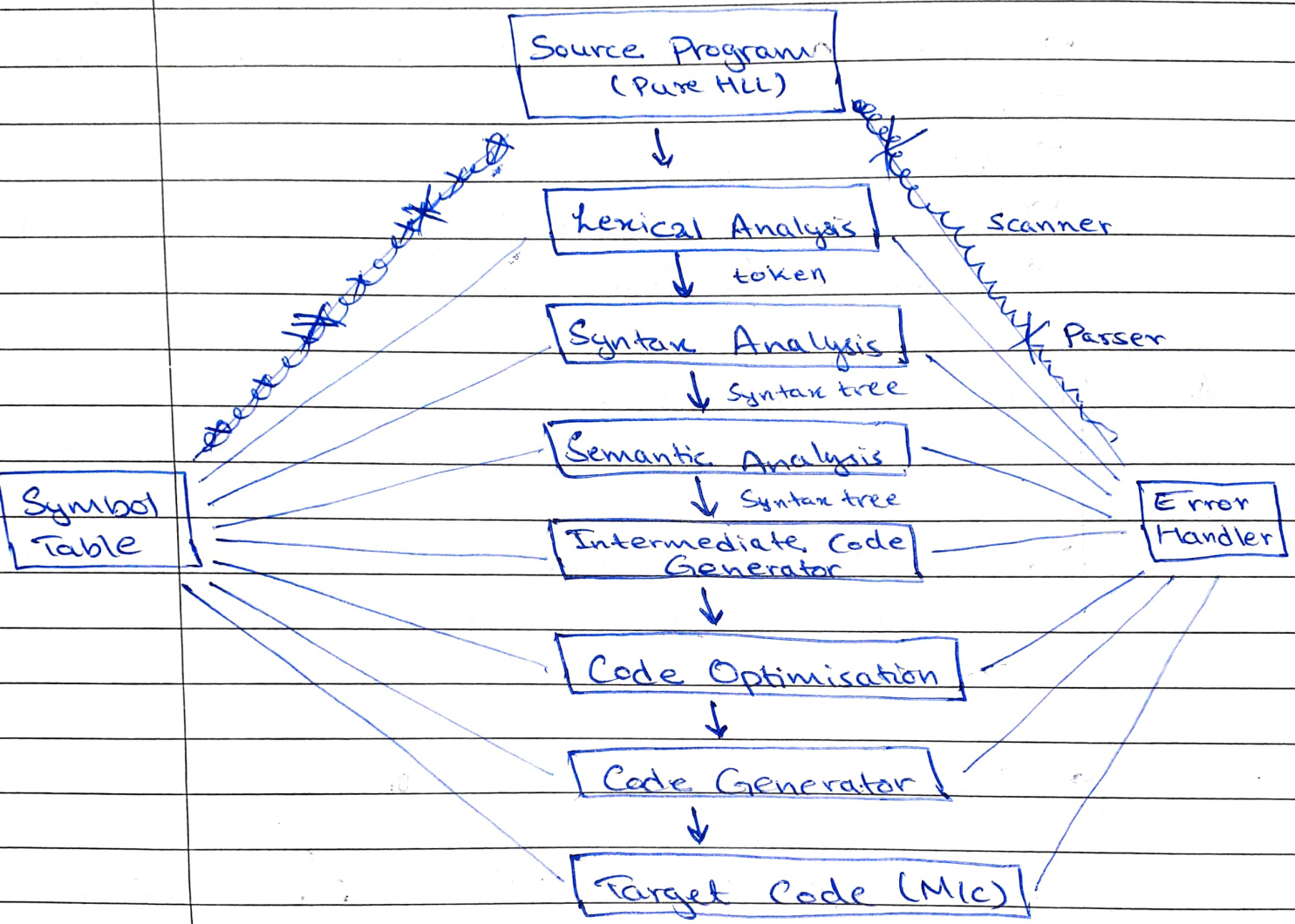
fig : Translation and Execution Process

- Formal Translation Model : DFA, NFA, PDA, Grammar, RE

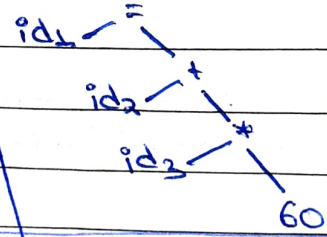
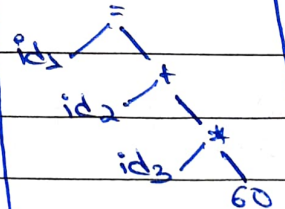
__/__/____

Stages in translation:

fig.: Stages in Compiler:



Example: $a = b + c * 60$

Lexical Analysis	Syntax Analysis	Semantic Analysis	Intermediate Code
$id_1, OP_1, id_2,$			$t_1 = 60.000000$
$OP_2, id_3,$			$t_2 = id_3 * t_1$
OP_3, construct			$t_3 = id_2 + t_2$
			$id_1 = t_3$

Code Optimisation	Code Generator
$t_1 = id_3 * 60.000000$	MOV id_3, R_2
$id = id_2 + t_1$	MUL #60.000000, R_2
	MOV id_2, R_1
	ADD R_2, R_1
	MOV R_1, id_1

__/__/____

Language Properties:

- Modelling Language Properties
  - Formal Properties of Language
    - Chomsky Hierarchy
    - Undecidability
  - Language Semantics
- Elemantary data types.
- Encaplusation.
- Inheritance
- Sequence Control
- Subprogram Control

• Chomsky Hierarchy:

Grammar Type	Grammar Accepted	Language Accepted	Machine
Type 0	Unrestricted Grammar	Reg. Enumerable Lang	Turing Mac.
Type 1	Context Sensitive Gr.	Context Sensitive Lang	Linear Bounded Automata
Type 2	Context free Gra.	Context free Lang.	Pushdown Automata
Type 3	Regular Grammar	Regular Language	Finite state Automata

• Undecidability: → Halting Program
→ Ambiguity.

• Language Semantics:

Production Rule	Semantic Action
$S \rightarrow E;$	Print (E.val)
$E \rightarrow E_1 + T$	E.val = E_1.val + T.val
$E \rightarrow T$	E.val = T.val
$T \rightarrow T_1 * F$	T.val = T_1.val * F.val
$T \rightarrow F$	T.val = f.val
$F \rightarrow$ digit	F.val = digit.lexical.

__/__/____

* ~~Language Properties~~
~~Programming Language Properties~~

- Elementary data type:
  - Properties of types and Data Object
    - Data Object
    - Variable
    - Constant
    - Data Type
    - Declaration
    - Type Checking
    - Type Conversion
  - Scalar Data Type
    - Numeric Data Type: int, float
    - Enumeration : enum
    - Boolean : bool
    - Character : char
  - Composite Data Type
    - Array
    - String
    - Pointer
    - File

__/__/____

PPL Syllabus:

ch-1 * Language Design Issues.
- Why study Programming Languages?
- Short History of Programming Languages.
- Role of Programming Languages.
  - Attributes of a Good Language.
  - Language Paradigms

ch-5 * Elementary Data Types.
- Data Objects, Variables and Constants.
- Scalar Data Types: Integers, Subranges, Floating Point Real Numbers, Fixed-Point Real Numbers, Complex Num., Rational Num., Enumerations, Booleans, Characters.
- Composite Data Types: Character Strings, Pointers and Programmer Constructed Data Objects, Files and Input-Output

ch-6 * Encapsulation.
- Structured Data Types: Vectors, Arrays, Records, Lists, Sets.
- Abstract Data Types: Information Hiding.
- Encapsulation by Subprograms.

ch-8 * Sequence Control
- Implicit and Explicit Sequence Control.
- Sequence with Arithmetic Expressions.
  - Tree Structured Representation
  - Execution-Time Representation
- Sequence Control Between Statements.

__/__/____

Ch-9 ✱ Subprogram Control

343 - Simple Call-Return Subprograms,

353 Recursive Subprograms.

356 - Names and Referencing Environments,

363 Static and Dynamic Scope.

374 - Parameter Transmission

Ch-10 ✱ Storage Management

416 - Elements Requiring Storage.

413 - Programmer and System Controlled Storage.

419 - Static Storage Management.

420 - Heap Storage Management.

422 • Fixed-Size Elements

425 • Variable-Size Elements.