SYSTEM PROGRAMMING

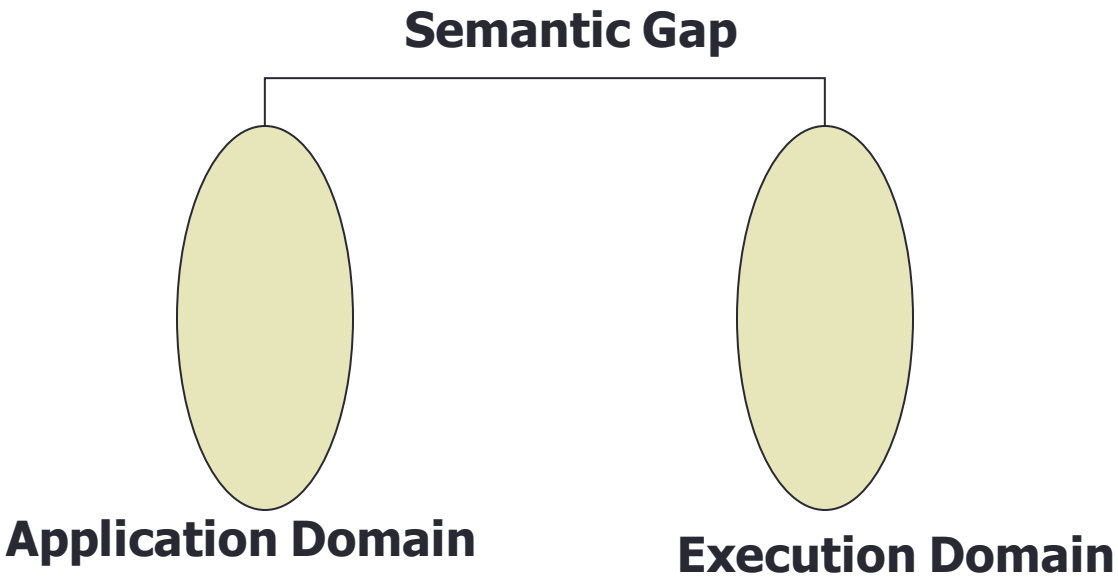
Lecture 1

Introduction – 1/9

- Why does language processing activities arise?
  - Software designer describes the ideas concerning behavior of s/w and how these ideas are implemented in a computer system
- Designer express ideas in terms related to the application domain of the s/w
- To implement these ideas, their description has to be interpreted in terms related to execution domain of computer system

Introduction – 2/9

- Terms :
  - **Semantics** represent rules of meaning of a domain
  - **Semantic gap** represent the difference between semantics of two domains



Introduction – 3/9

- The semantic gap has consequences:
  - large development times
  - large development efforts
  - poor quality of s/w
- These issues are tackled by s/w engineering through use of programming languages (PLs)

Introduction – 4/9

- S/w engineering steps aimed at the use of a PL can be grouped into
  - Specification, design and coding steps
  - PL implementation steps
- S/w implementation using a PL introduces a new domain, the **PL domain**
- Semantic gap between the application domain and execution domain is bridged by s/w eng steps

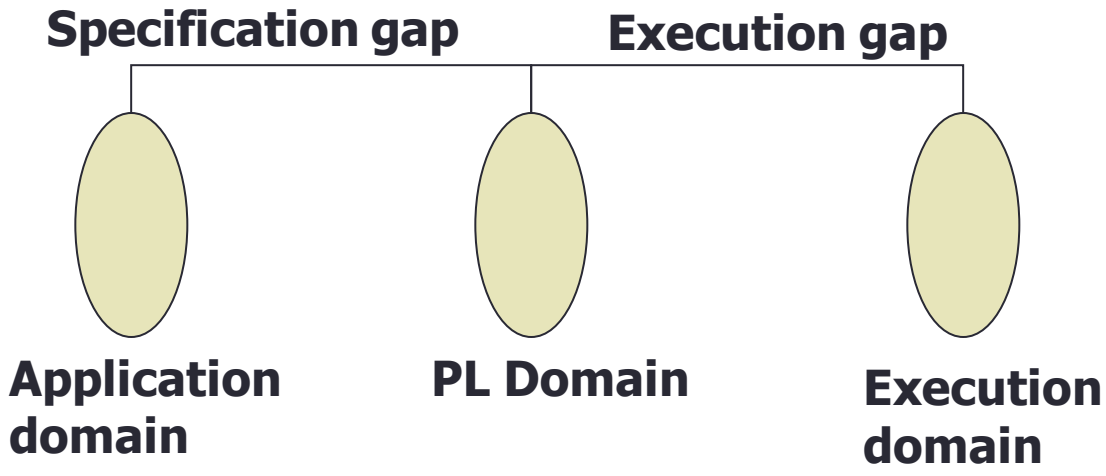
Introduction – 5/9

- First step → bridges gap between application and PL domain
- Second step → bridges gap between PL and execution domain
- Terms :
  - **Specification gap** : Gap between application and PL domain
  - **Execution gap** : Gap between PL and execution domain

Introduction – 6/9

- **Specification gap** → bridged by the s/w development team

- **Execution gap** → bridged by designer of PL processor, via a translator or interpreter



Introduction – 7/9

- Advantages of introducing PL domain
  - **Reduces severity of consequences :** Gap to be bridged by s/w designer is now between application domain and PL domain rather than between application and execution domain.
  - **Improving quality of s/w :** Language processor provides a diagnostic capability which detects and indicates errors in its i/p.

Introduction – 8/9

- **Specification gap** :- It is semantic gap between two specifications of same task
- **Execution gap** :- It is gap between semantics of programs (that perform same task) written in different programming languages
- Each domain has a **specification language (SL)**
- A specification written is an SL is a program in SL

Introduction – 9/9

- **Specification language of PL domain is the PL itself**

- **Specification language of execution domain is machine language of computer system**

Language Processors – 1/5

- **A language processor is a s/w which bridges a specification or execution gap**
- **Language processing** - describe the activity performed by a language processor
- **Source program** : Program form input to a language processor as the source program
- **Target program** : Program form output from a language processor

Language Processors – 2/5

- **Source language** : Language in which source programs are written
- **Target language** : Language in which target programs are written
- Language processor typically abandons generation of target program if it detects errors in source program

Language Processors – 3/5

- A language translator bridges an execution gap to machine language (or assembly language) of computer system.
  - An **assembler** is a language translator whose source language is assembly language.
  - A **compiler** is any language translator which is not an assembler
- A **detranslator** bridges the same execution gap as language translator, but in reverse direction

Language Processors – 4/5

- A **preprocessor** is a language processor which bridges an execution gap but is not a language translator

Language Processors – 5/5

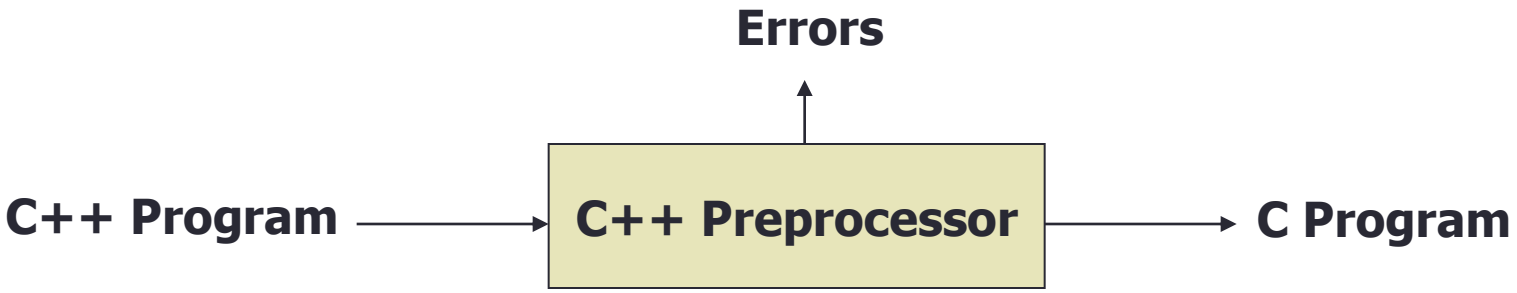


Fig (a) above is a preprocessor as it converts one program into another

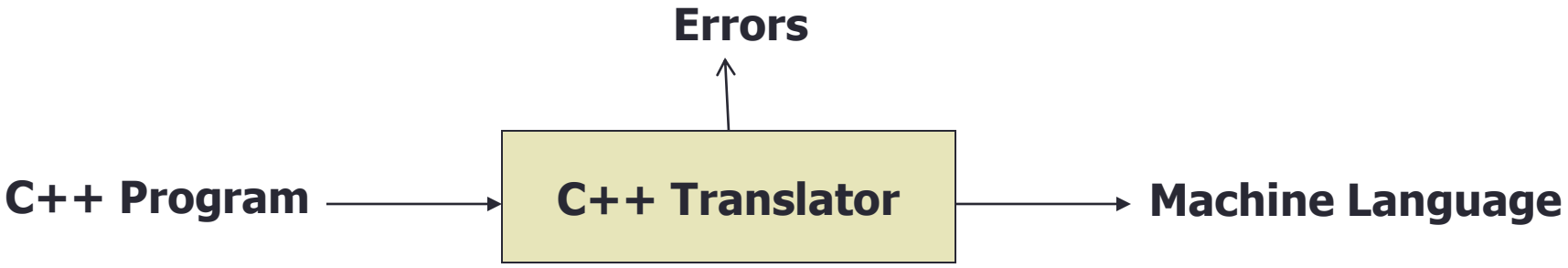


Fig (b) above is a translator since it produces a machine language program

Interpreters – 1/2

- **An interpreter is a language processor which bridges an execution gap w/o generating a machine language program**
- An Interpreter ‘executes’ a program written in a PL
- Execution gap vanishes totally
- The specification language of PL domain is identical with that of interpreter domain

Interpreters – 2/2

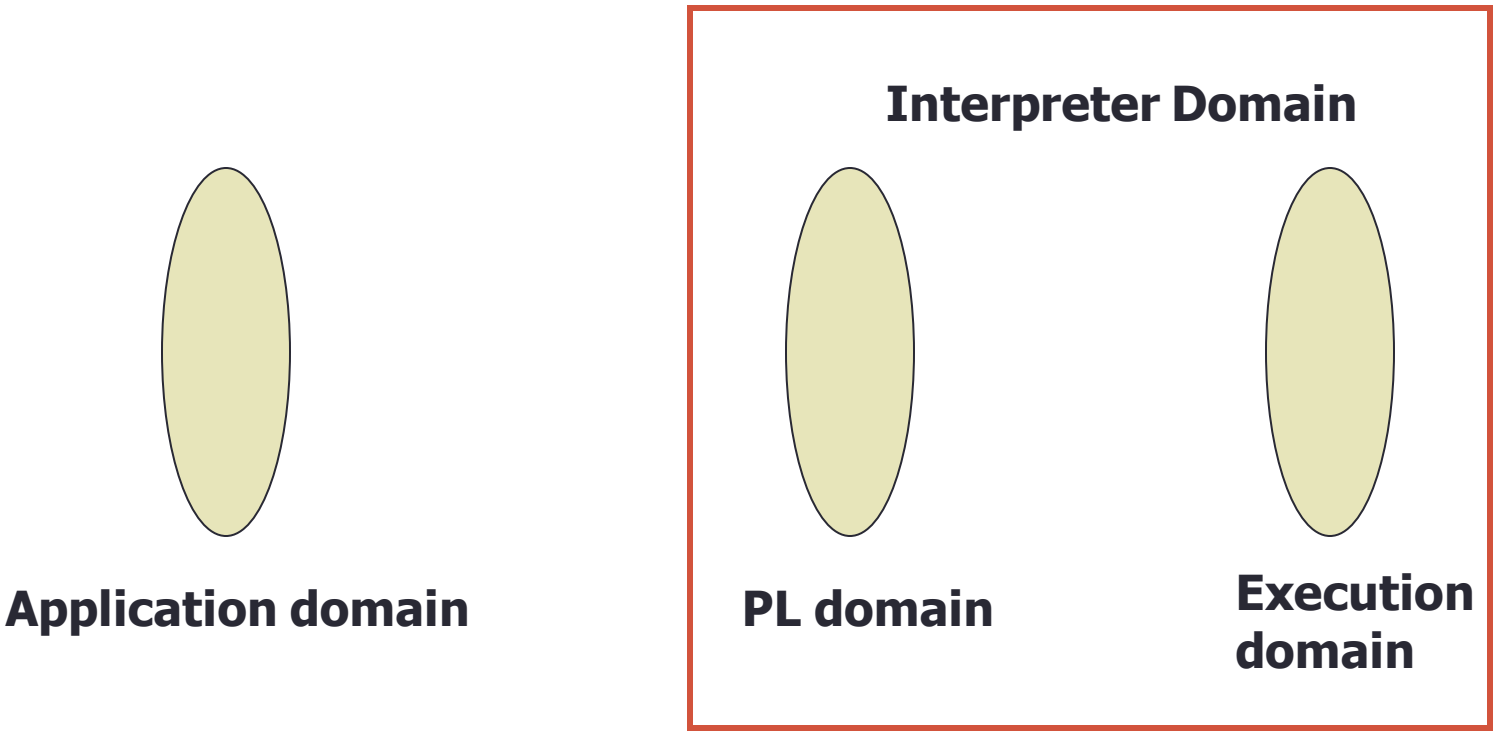


Fig : Schematic representation of an interpreter, wherein interpreter encompasses the PL domain as well as execution domain

Problem-oriented & Procedure-oriented languages – 1/2

Problem-oriented languages

- Used for specific applications
- Have large execution gaps
- Gap is bridged by translator or interpreter and does not concern s/w

Procedure-oriented languages

- Provides general purpose facilities required in most application domains
- Independent of specific application domains
- Results in large specification gap which has to be bridged by an application designer

Problem-oriented & Procedure-oriented languages – 2/2

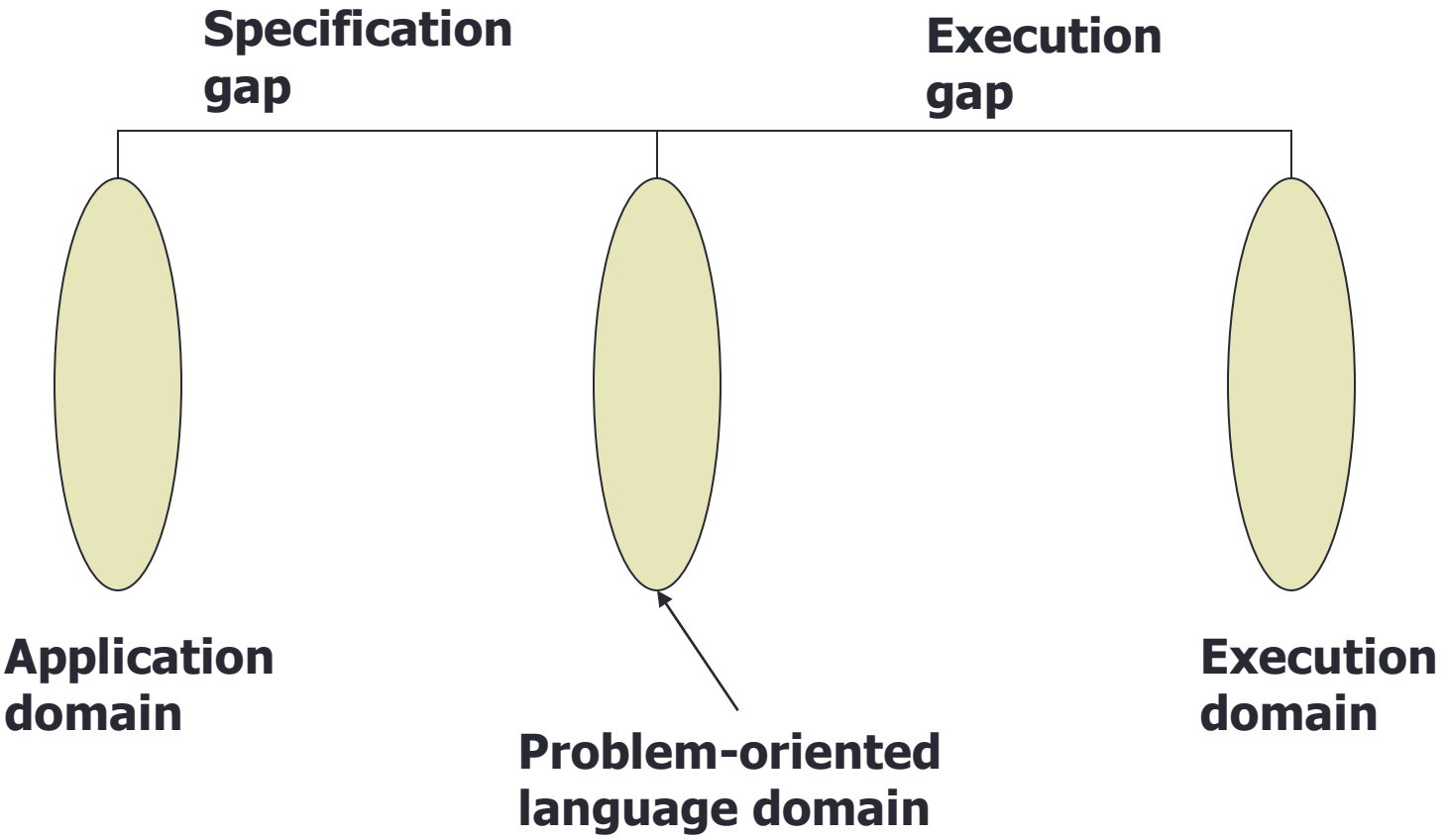


Fig: Problem-oriented language domain