

## 2.4 不同抽象层级的计算机（软件）

哈尔滨工业大学（深圳）  
计算机学院

# 主要内容

## 1.人机交互层面的计算机：计算机语言与编译器

汇编语言与汇编器

高级语言与编译器

不同层级语言与编译器/虚拟机

## 2.机机交互层面的计算机：协议与编解码器

图像表示

声音表示

视频表示

协议与编解码器

## **2.4 .1 人机交互层面的计算机---计 算机语言与编译器**

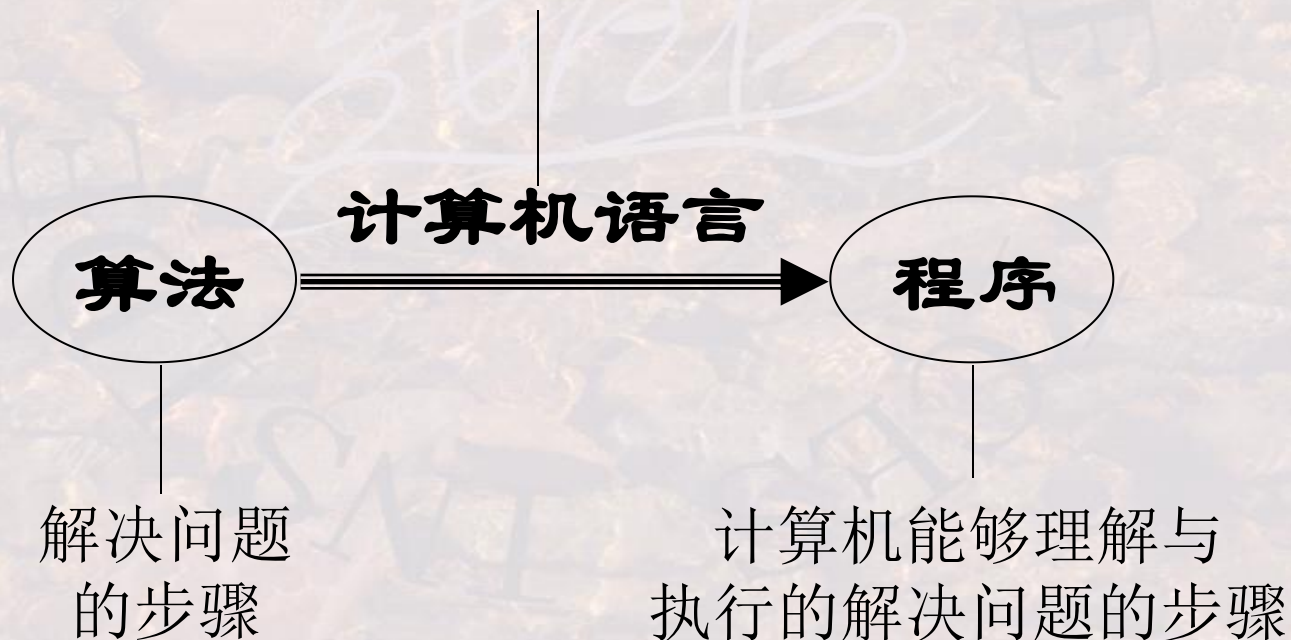


# 1. 汇编语言与编译器（汇编程序）

## (1) 为什么需要计算机语言？

### 算法、计算机语言与计算机程序

步骤书写的规范、语法规则、标准的集合  
是人和计算机都能理解的语言



# 1. 汇编语言与编译器（汇编程序）

## (2) 机器语言：计算机能够理解与执行的唯一语言

### 计算机语言---机器语言

#### 指令系统

**指令系统：** CPU可以解释并执行的二进制编码命令集合。

操作码

地址码

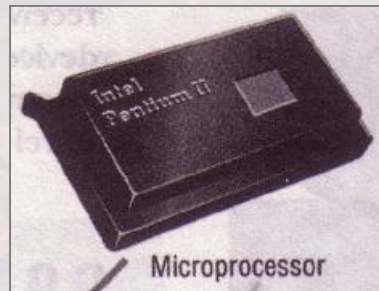
100001	10 00000111
100010	11 00001010

#### 机器语言

**机器语言：** 二进制和编码方式提供的指令系统可作为编写程序的语言，被称为机器语言

所有程序都需转换成机器语言程序，计算机才能执行

问：用机器语言编写程序存在什么问题呢？



计算7+10并存储的程序

**100001 10**  
**00000111**

**100010 10**  
**00001010**

**100101 11**  
**00000110**

**111101 00**

# 1. 汇编语言与编译器（汇编程序）

## (3) 汇编语言与编译器

### 计算机语言---汇编语言

◆用符号编写程序 ==> 翻译 ==> 机器语言程序

◆人们提供了用助记符编写程序的规范/标准。同时开发了一个翻译程序，实现了将符号程序自动转换成机器语言程序的功能。



操作码	地址码
100001	1000000111

↓

MOV A, 7

计算7+10并存储的程序

```
MOV A, 7
ADD A, 10
MOV (6), A
HLT
```

◆汇编语言：是用助记符编写程序的语言。

◆汇编语言源程序：是用汇编语言编出的程序。

◆汇编程序：是将汇编语言源程序翻译成机器语言程序的程序。

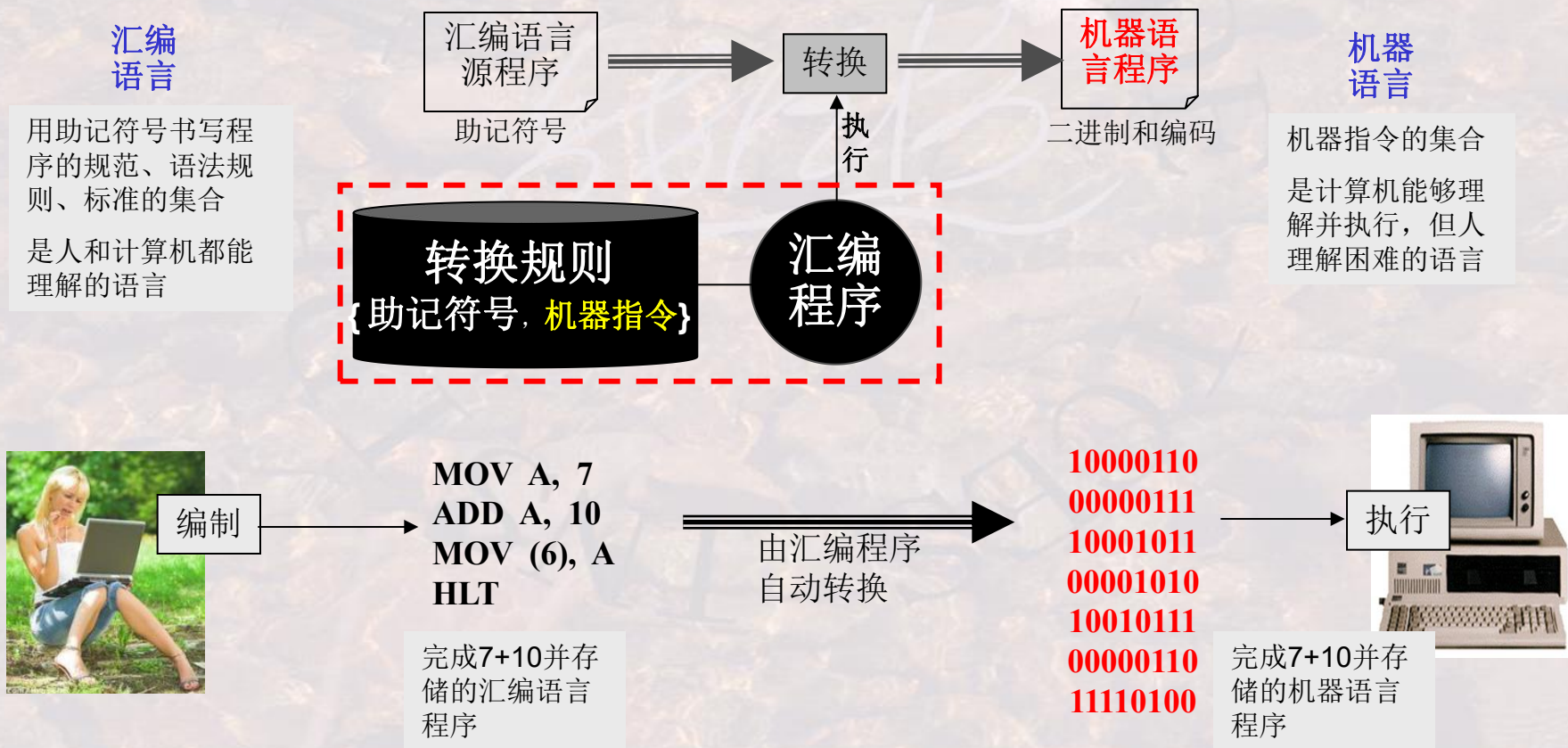


# 1. 汇编语言与编译器 (汇编程序)

## (4)汇编语言与编译器

### 计算机语言---汇编语言---汇编程序(编译器)

#### ◆汇编语言程序处理过程



## 2. 高级语言与编译器

### (1)为什么还要提出高级语言？

#### 计算机语言---高级语言

◆人们提供了类似于**自然语言方式**、以**语句为单位**书写程序的**规范/标准**。并开发了一个**翻译程序**，实现了将语句程序**自动**翻译成机器语言程序的功能。

- ◆**高级语言**：是用类似自然语言的语句编写程序的语言。
- ◆**高级语言源程序**：是用高级语言编出的程序。
- ◆**编译程序**：是将高级语言源程序翻译成机器语言程序的程序。



计算7+10并存储的程序

```
Result = 7+10;  
Return
```



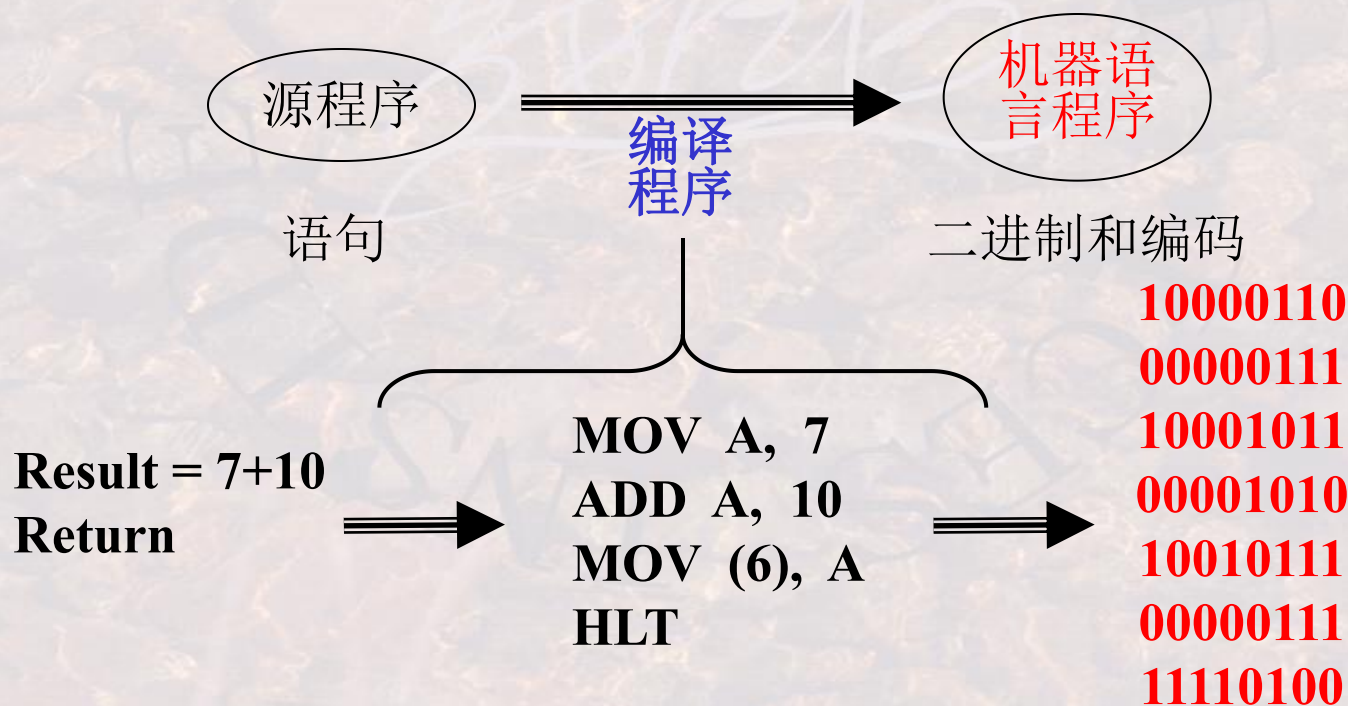
## 2. 高级语言与编译器

### (2) 高级语言和汇编语言的差别在哪里？

◆ **高级语言**：机器无关性；一条高级语言语句往往可由若干条机器语言语句实现且不具有对应性

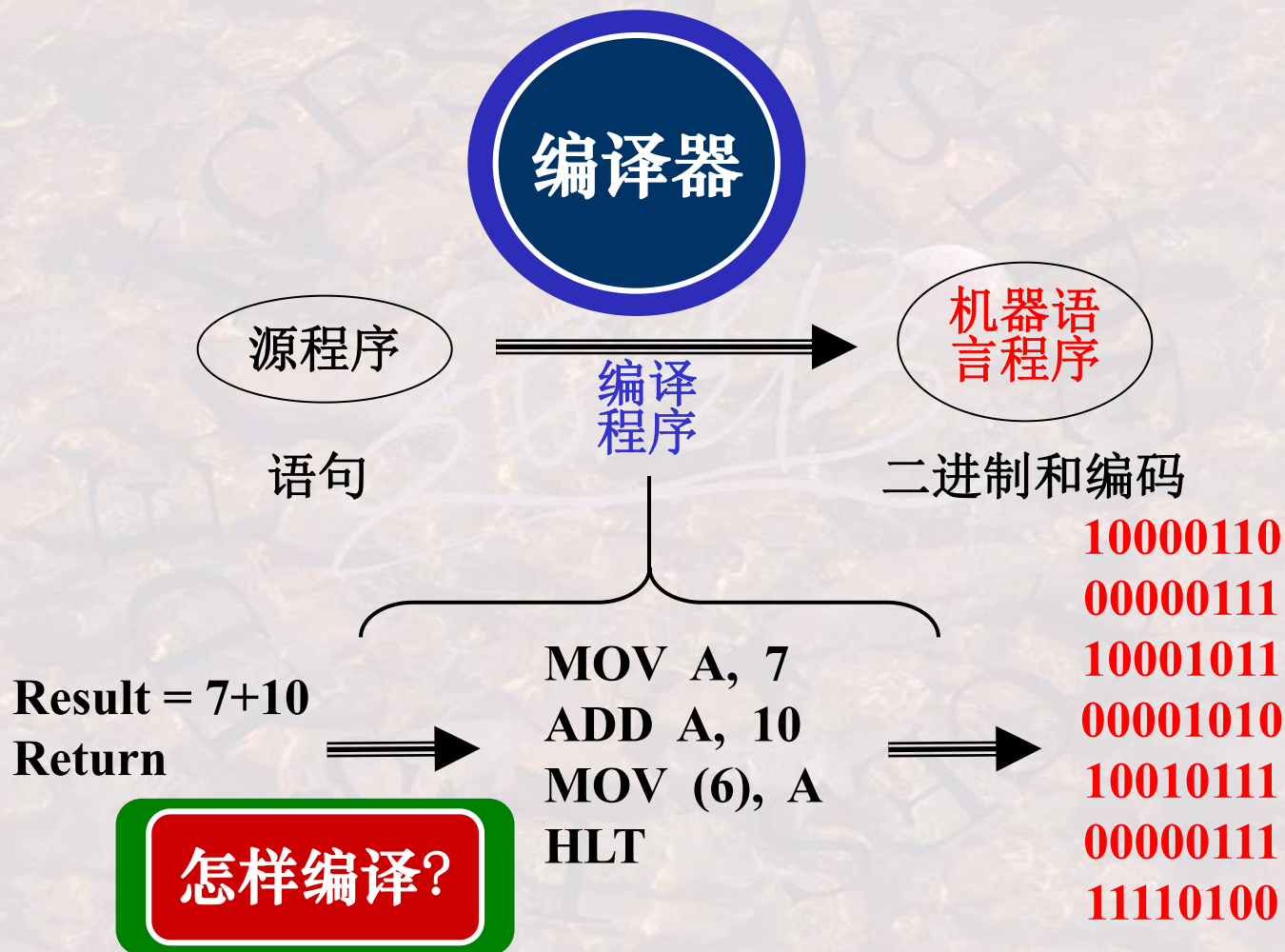
◆ **汇编语言**：机器相关性；汇编语言语句和机器语言语句有对应性

### 高级语言程序处理过程示意



## 2. 高级语言与编译器

### (3)为什么高级语言程序需要编译?



## 2. 高级语言与编译器

### (4)赋值语句编译示例：模式化的语句

由“**具体的**”运算式到“**模式**”运算式 (词法分析)

**Result = 7 + 10;**

**Sum = 8 + 15;**

**K = 100 + 105;**

**....**

**V = C + C;**

注：

**Result:** 具体的变量  
**7, 10:** 具体的

变化的部分

注：

**V:** 变量  
**C:** 常量

**=** 赋值符号  
**+** 加法运算符号  
**;** 语句结束符

不变的部分  
(保留字)

**=** 赋值符号  
**+** 加法运算符号  
**;** 语句结束符



## 2. 高级语言与编译器

### (5)赋值语句编译示例：语句模式的识别

“模式” 运算式的识别及常量、变量的标识（语法分析）

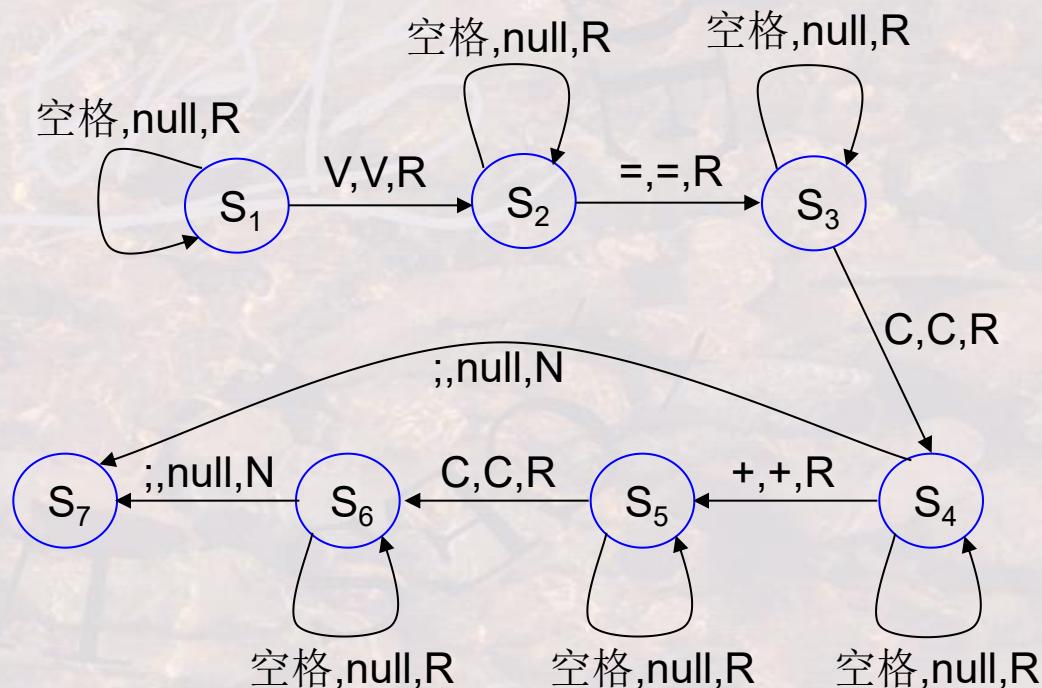
**V = C + C;**

注：字母表{V, C, =, +, 空格, ;}; S<sub>1</sub>起始状态; S<sub>7</sub>终止状态; null表示什么也不写回。

**Result = 7 + 10;**



**(V, 1) = (C, 1) + (C, 2);**



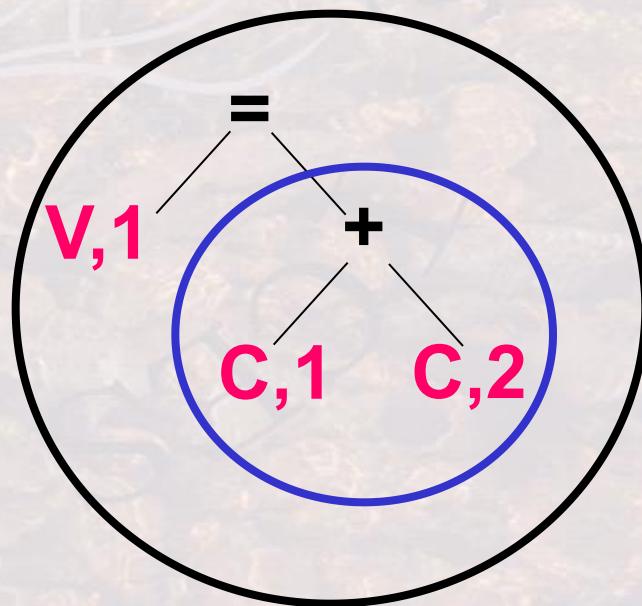
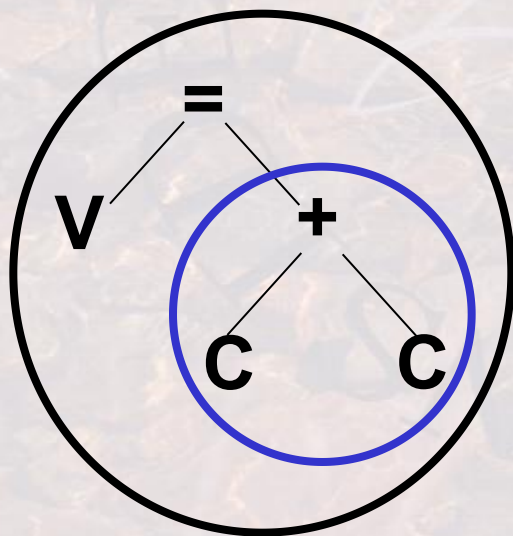
(c)能识别两种模式 “V=C;”和  
“V=C+C;”并能去除空格的图灵机示  
意图

## 2. 高级语言与编译器

### (6)赋值语句编译示例：复杂模式的预先构造

复杂模式转换为简单模式及其组合（语法分析）

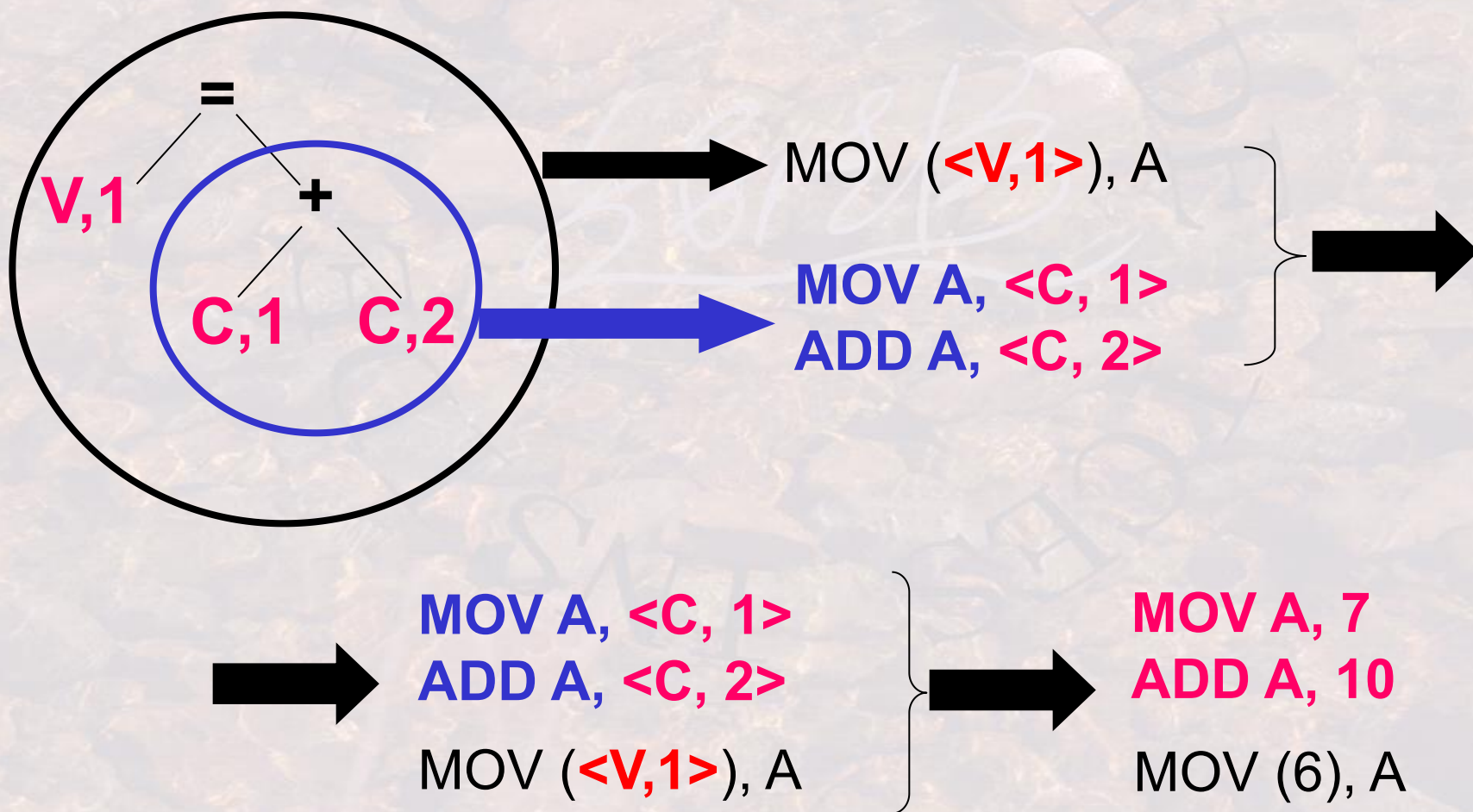
**V = C + C;**



## 2. 高级语言与编译器

### (7)赋值语句编译示例：简单模式与汇编语句的映射

将简单模式转换成汇编语言语句序列，用常量值和变量地址进行替换，组合次序调整，得到最后的汇编语言程序（语义分析）

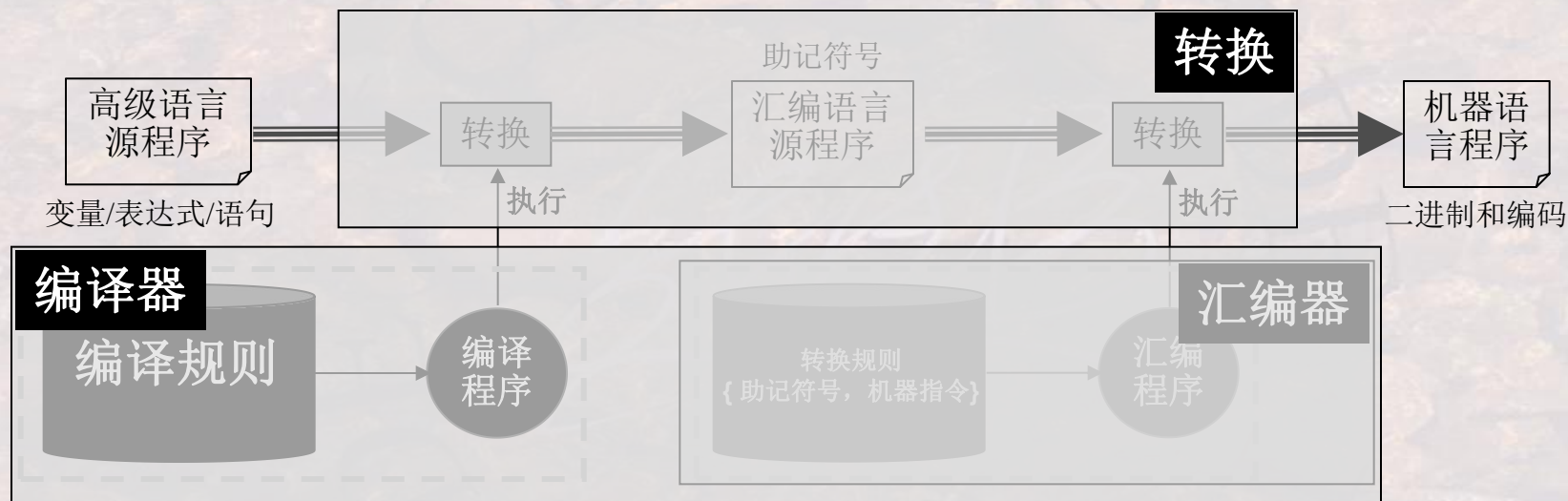




## 2. 高级语言与编译器

### (8)高级语言编译器如何实现呢？

#### 高级语言编译器



高级语言

编制

Result = 7+10  
Return

自动转换

转换

MOV A, 7  
ADD A, 10  
MOV (6), A  
HLT

自动转换

10000110  
00000111  
10001011  
00001010  
10010111  
00000110  
11110100

执行



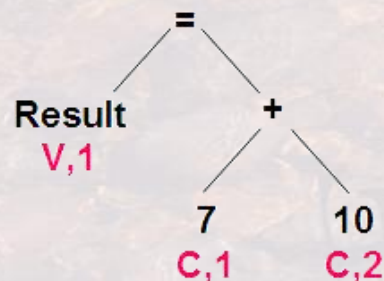
机器语言

## 2. 高级语言与编译器

### (9)小结

**Result = 7 + 10;**

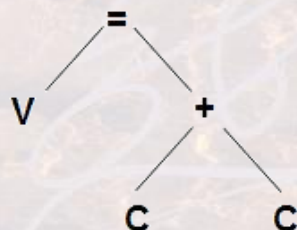
注：  
Result: 具体的变量  
7, 10: 具体的常量  
=: 赋值符号  
+: 加法运算符



(a)一种具体的语句  
及其解析结构

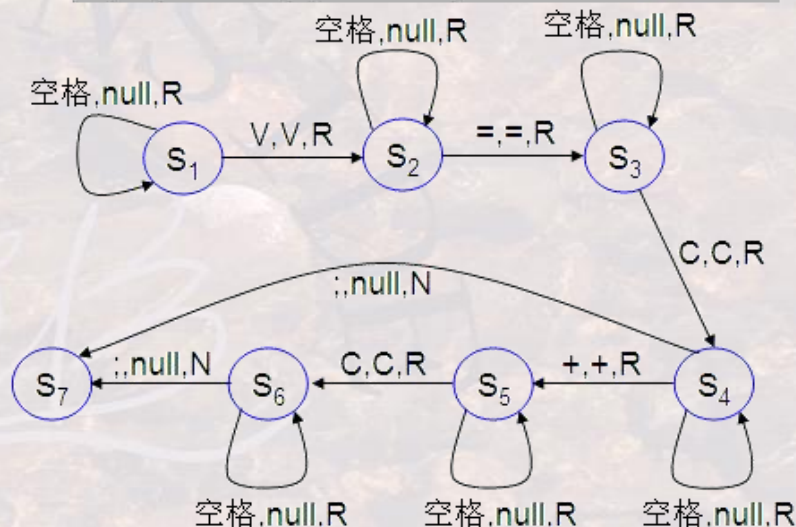
**V = C + C;**

注：  
V: 变量  
C: 常量  
=: 赋值符号  
+: 加法运算符

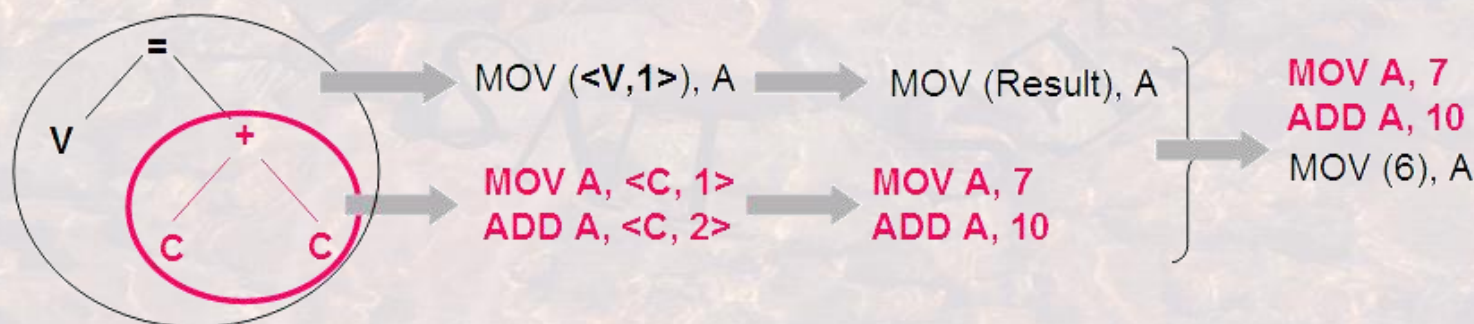


(b)图(a)所示语句的一种  
模式及其解析结构

注：字母表{V, C, =, +, 空格, ;}; S<sub>1</sub>起始状态; S<sub>7</sub>终止状态; null表示什么也不写回。



(c)能识别两种模式“V=C;”和“V=C+C;”  
并能去除空格的图灵机示意图



(d)语法分析树转换成汇编语言语句的过程示意

## (10)小结

## 语句词汇的识别(词法分析)

## 语句模式的识别(语法分析)

# 形式语言 与自动机

# 编译系统/ 编译原理

## 复杂语句模式的构造

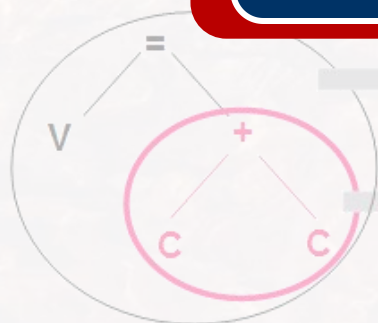
## 基本语句模式与汇编语句的映射

## 汇编语句的组 装与次序调整

# 常量与变量的替换



(a)一种具体的语言及其解析结构



MOV (<V,1>), A

```
MOV A, <C, 1>
ADD A, <C, 2>
```

MOV (Result), A

```
MOV A,
ADD A,
```

```
MOV A, 7
ADD A, 10
```

NAVJAG (6), A

(d) 语法分析树转换成汇编语言语句



关于计算机语言的编译，下列说法不正确的是\_\_\_\_\_。

- ☐ A 需要“分词”，将其中的常量、变量名和保留字识别出来，并分类及编号；
- ☐ B 需要识别每一条语句所对应的“模式”。任意语句的常量和变量名被归为“标识符”类别，而标识符与保留字的不同组合关系构成了语句的模式；计算机语言是由有限的语句模式构成的；
- ☐ C 对每一种模式，都有相应的组合构造方法，即模式可被认为是由原子模式或说基本模式通过组合的方法构造出来的，对原子模式或者基本模式可以事先写好其相应的目标语言的指令或语句；
- ☐ D 按照模式由原子模式的组合次序，可将模式语句转换成目标语言的指令或语句；进一步按照分类及编号将常量、变量名代入形成最终的目标语言程序，完成编译；
- ☒ E 上述有不正确的。

提交

### **3. 计算机语言与编译器—不同层 级语言与编译器/虚拟机器**

# 计算机语言的发展

## (1)如何更方便地编写程序？

### 用高级语言编写程序

符号化

语句化

结构化

```
K = 0;
```

```
For I = 1 to 100 Step 1
```

```
{ If I <= 50 && I > 30
```

```
{ K = K + I; }
```

```
}
```

高级语言源程序

编译化

机器语言程序

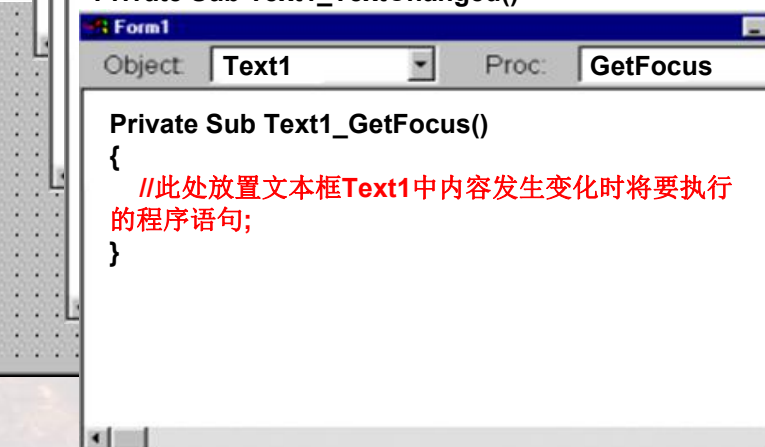
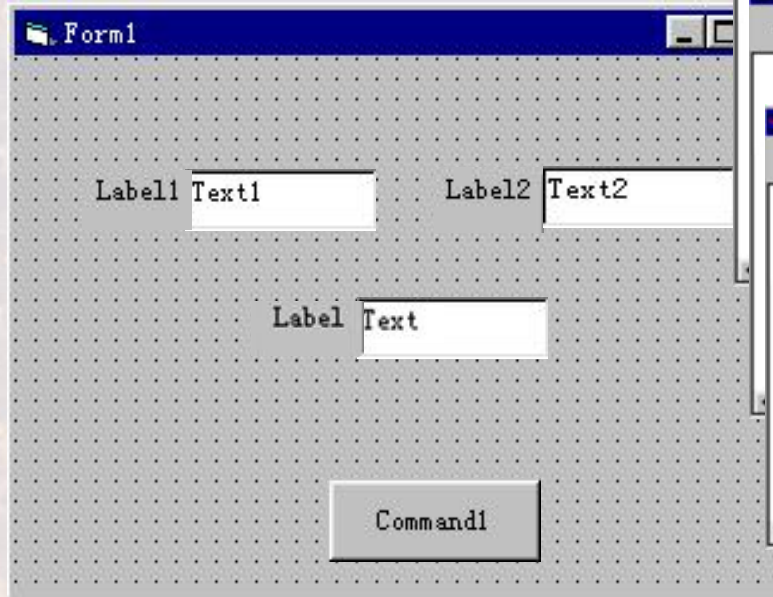


# 计算机语言的发展

## (2)如何更方便地编写程序？

### 面向对象的程序设计语言与 可视化构造语言

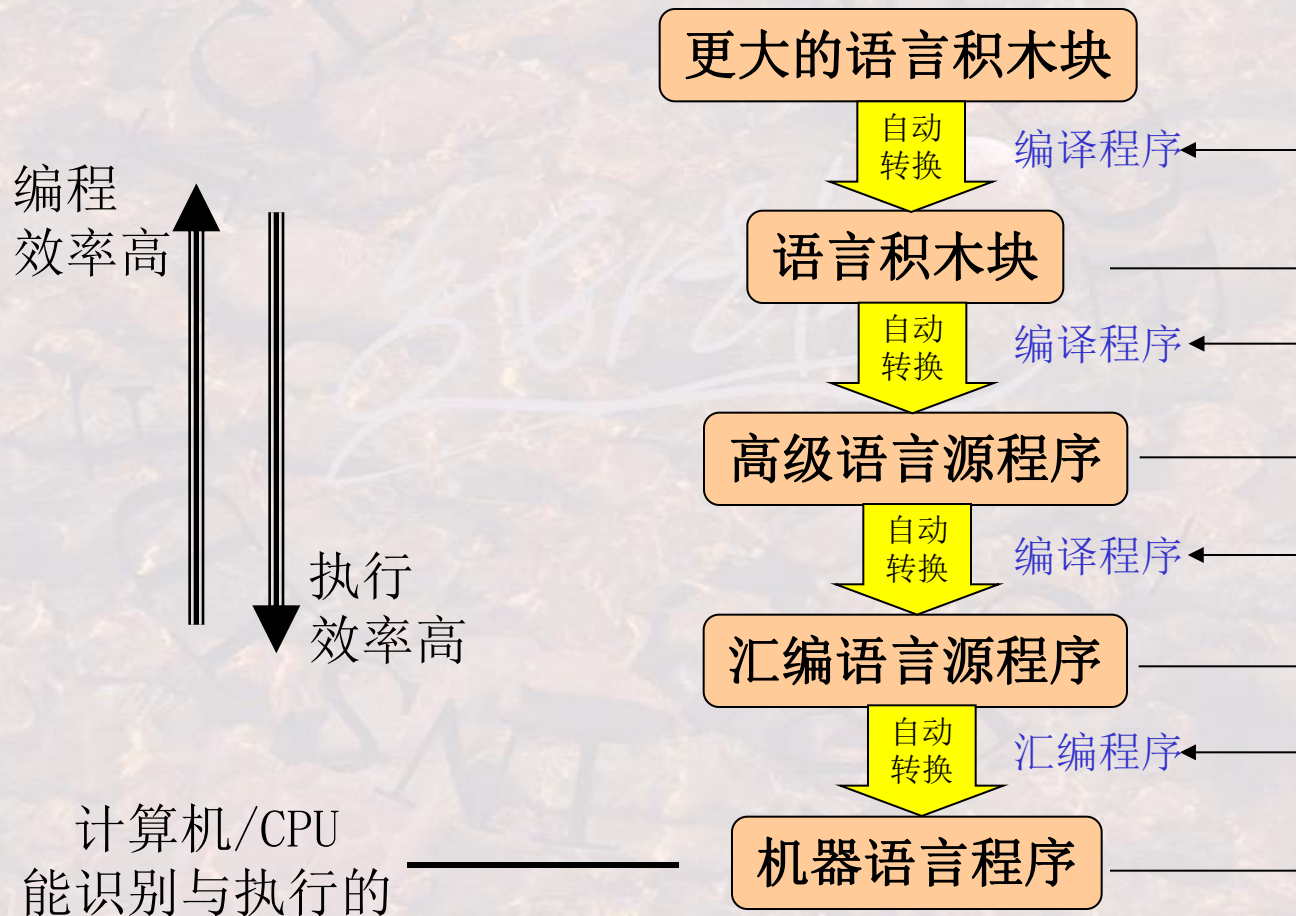
----像堆积木一样构造程序



# 计算机语言的发展

## (3) 计算机语言的发展思维?

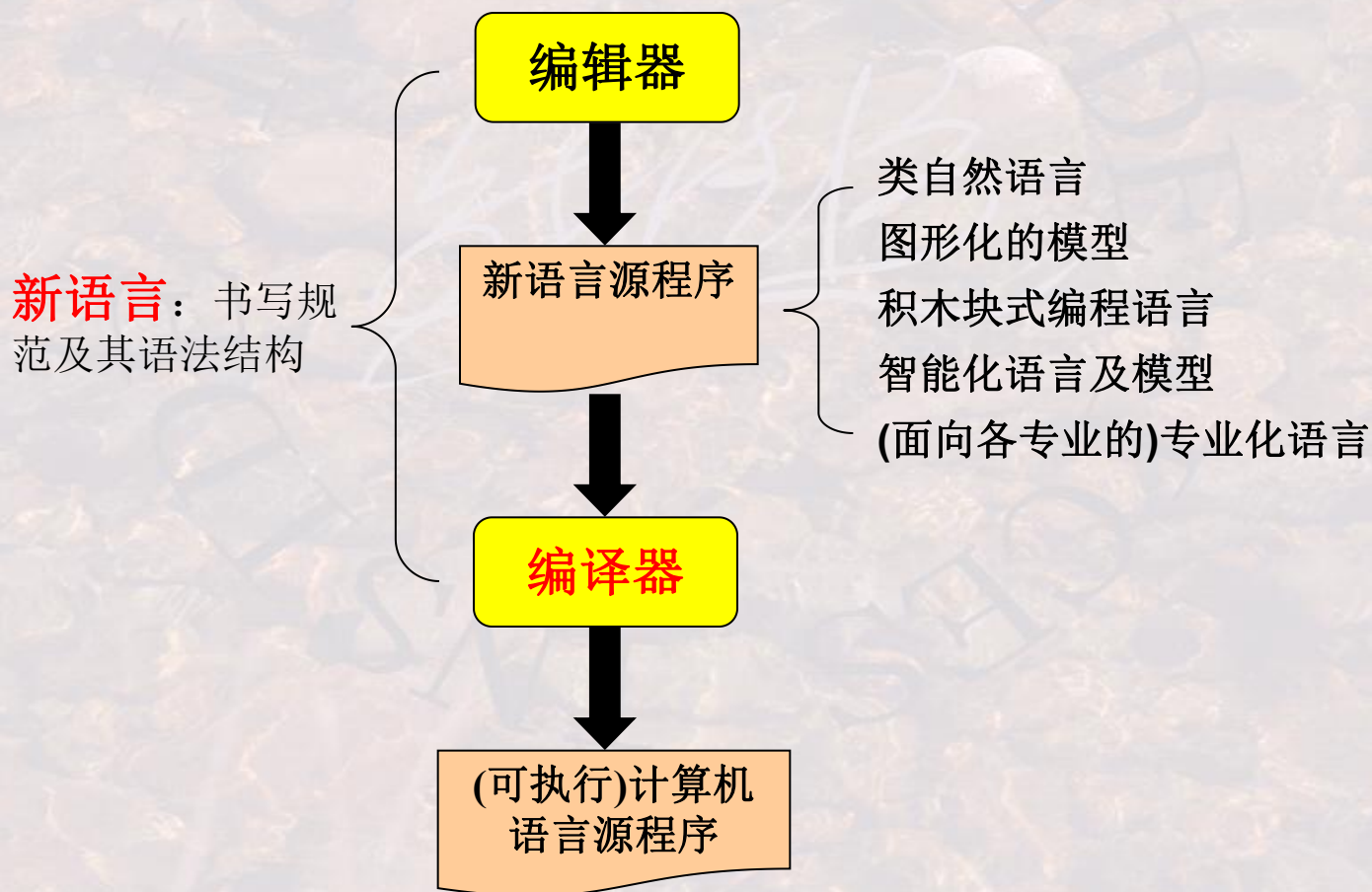
### 计算机语言发展的基本思维



# 计算机语言的发展

## (4)能否提出新语言?

不仅要用语言，还要发明新语言





# 计算机语言的发展

## (5)能否提出新语言？

计算机技术是伴随着计算机语言的不断发展而发展起来的

### ◆因计算机语言获得图灵奖的

- 1966 A.J. Perlis: 编程技术和编译架构
- 1972 E.W. Dijkstra: **ALGOL**语言
- 1974 Donald E. Knuth: 程序语言
- 1977 John Backus : 高级语言, **Fortran**
- 1979 Kenneth E. Iverson: 编程语言, **APL**
- 1980 C. Antony R. Hoare: 编程语言
- 1981 Edgar F. Codd: 关系数据库语言
- 1984 Niklaus Wirth: 开发了**EULER**、**ALGOL-W**、**MODULA**和**PASCAL**一系列崭新的计算语言。
- 1987 John Cocke: 编译器
- 2001 Ole-Johan Dahl、Kristen Nygaard: 面向对象编程,**SIMULA I** 和 **SIMULA 67**中。
- 2003 Alan Kay :面向对象语言, **Smalltalk**
- 2005 Peter Naur:**Algol60**程序语言。
- 2006 Fran Allen: 编译器

关于计算机语言，下列说法正确的是

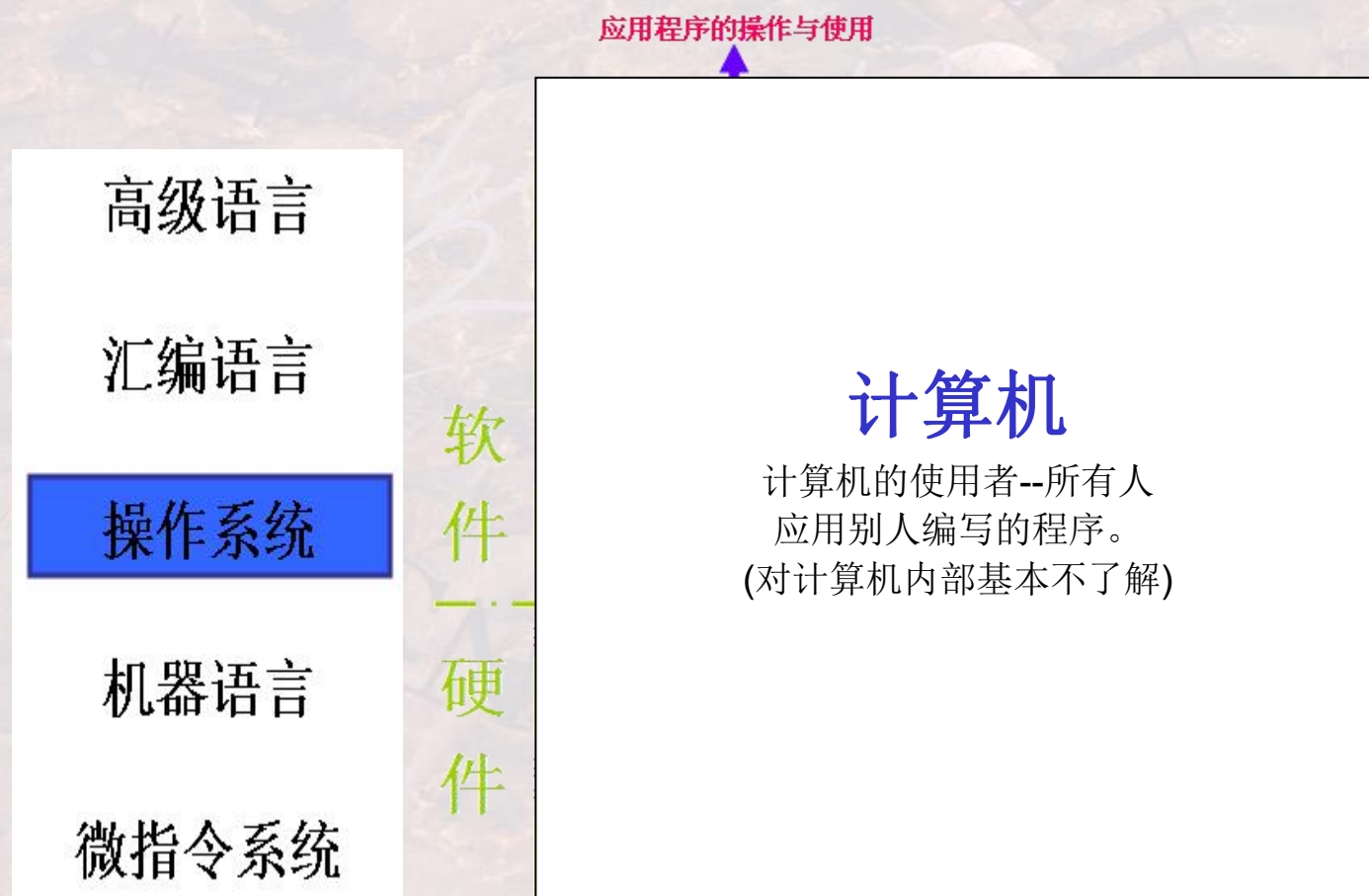
- ☒ A 所有源程序最后都需被转换为机器语言程序，机器才能够执行；
- ☒ B 所谓“高级语言”和“低级语言”是指其和机器硬件的相关程度，不涉及机器硬件的语言为高级语言，而与机器硬件相关的语言则为低级语言；
- ☒ C 低级语言程序执行效率高是因为用低级语言编程时可以充分利用硬件的各种特殊性，而高级语言则只能使用硬件的标准结构；
- ☒ D 高级语言编程效率高是因为其可用大粒度积木块来构造程序，比一行行语句、一条条指令来编程效率高出很多。

提交

# 不同抽象层级计算机(虚拟机器)

## (1)你只想简单使用计算机吗？

计算机语言促进了计算机处理能力的不断增强





# 不同抽象层级计算机(虚拟机器)

## (2)你想编程序但又不涉及硬件吗？

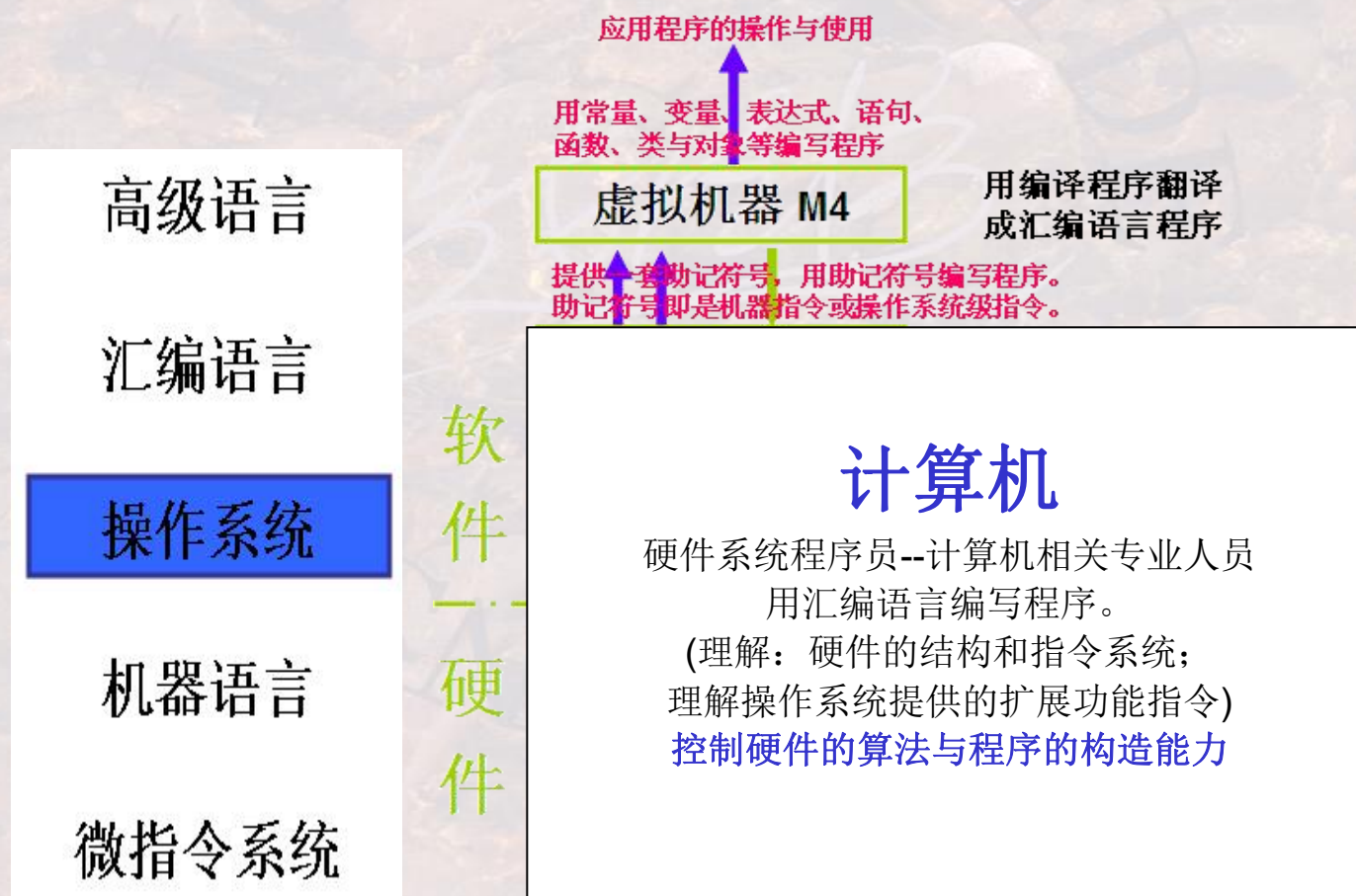
计算机语言促进了计算机处理能力的不断增强



# 不同抽象层级计算机(虚拟机器)

## (3)你想控制硬件设备吗?

### 计算机语言促进了计算机处理能力的不断增强



# 不同抽象层级计算机(虚拟机器)

## (4)你想编一些操作系统才能做的程序吗？

### 计算机语言促进了计算机处理能力的不断增强





# 不同抽象层级计算机(虚拟机器)

## (5)你想做芯片级的应用开发工作吗?

计算机语言促进了计算机处理能力的不断增强



# 不同抽象层级计算机(虚拟机器)

## (6)你想设计计算机相关的各种芯片和电路吗?

计算机语言促进了计算机处理能力的不断增强

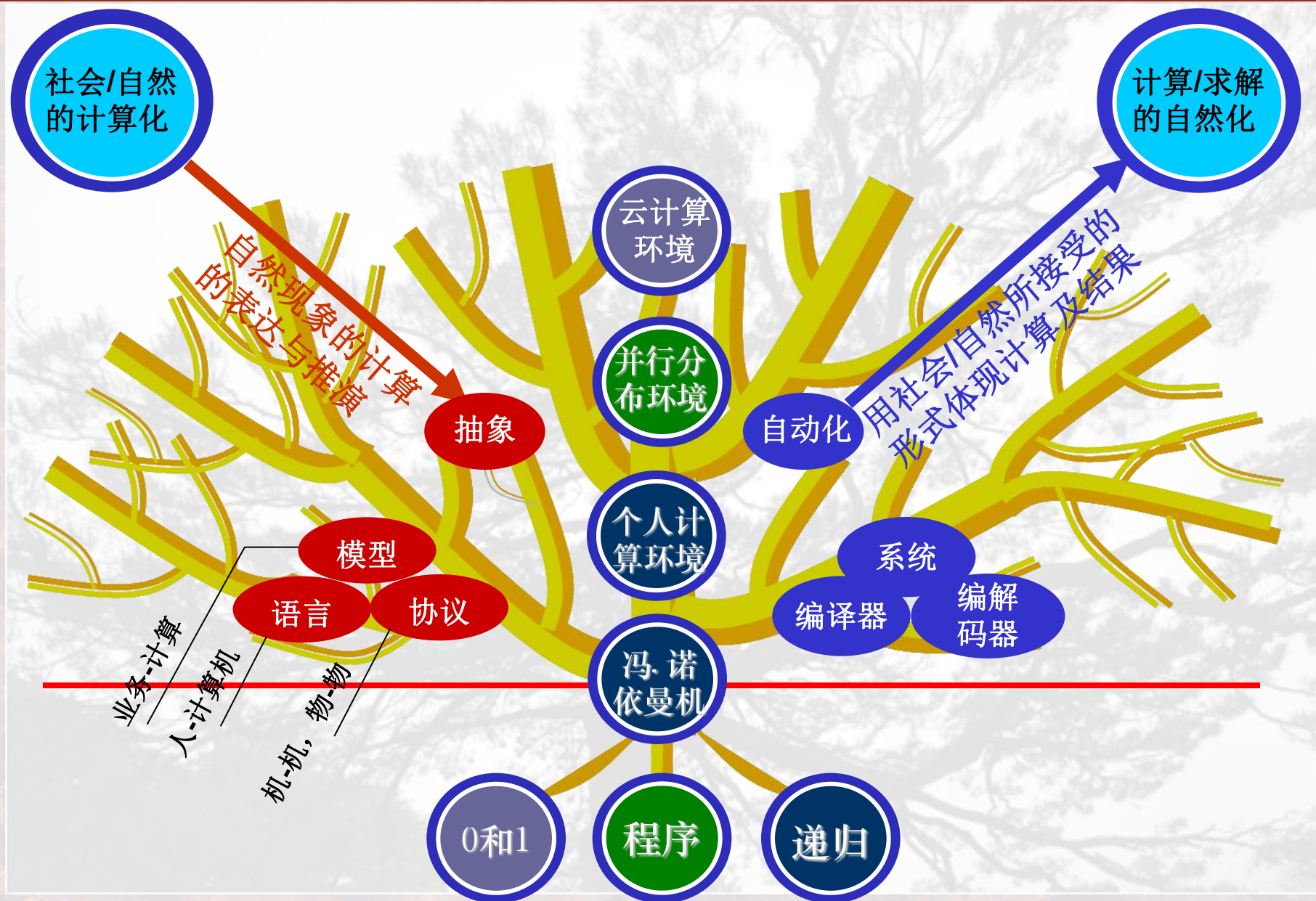


## **2.4.2 机-机交互层级的计算机：协议 与编码器/解码器/转换器/处理器**



# 协议与编解码器--另一种抽象-自动化机制示例

## (1)你记得计算之树中的不同抽象层次吗？



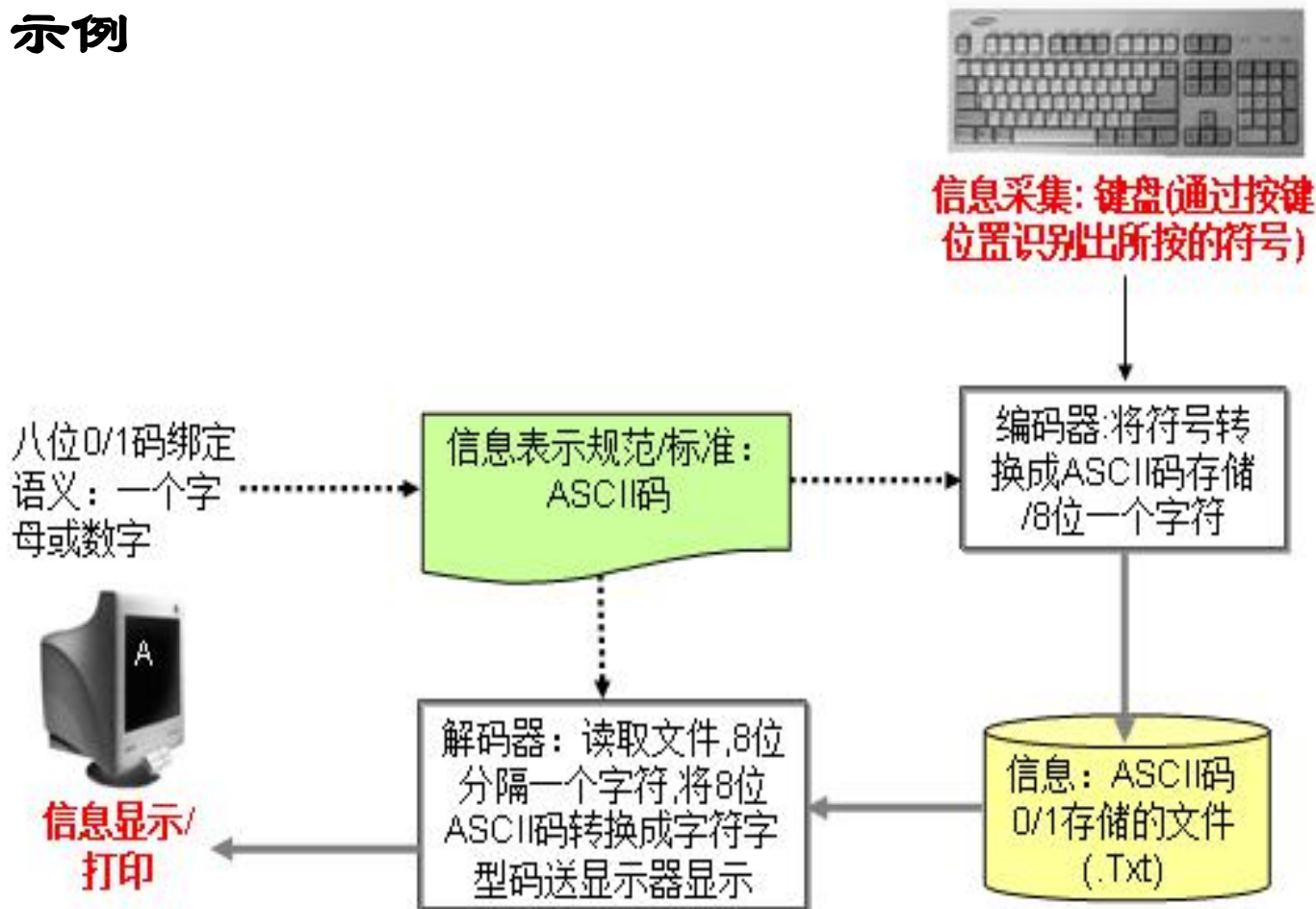
# 协议与编解码器--另一种抽象-自动化机制示例

## (2)怎样解决机器与机器之间的交互问题？

### 协议及其应用示意： 键盘输入/与显示器显示处理

一般而言，  
“协议”是为  
交流信息的双方(计算机)能够正确实现信息交流而建立的一套规则、标准或约定。

#### 示例



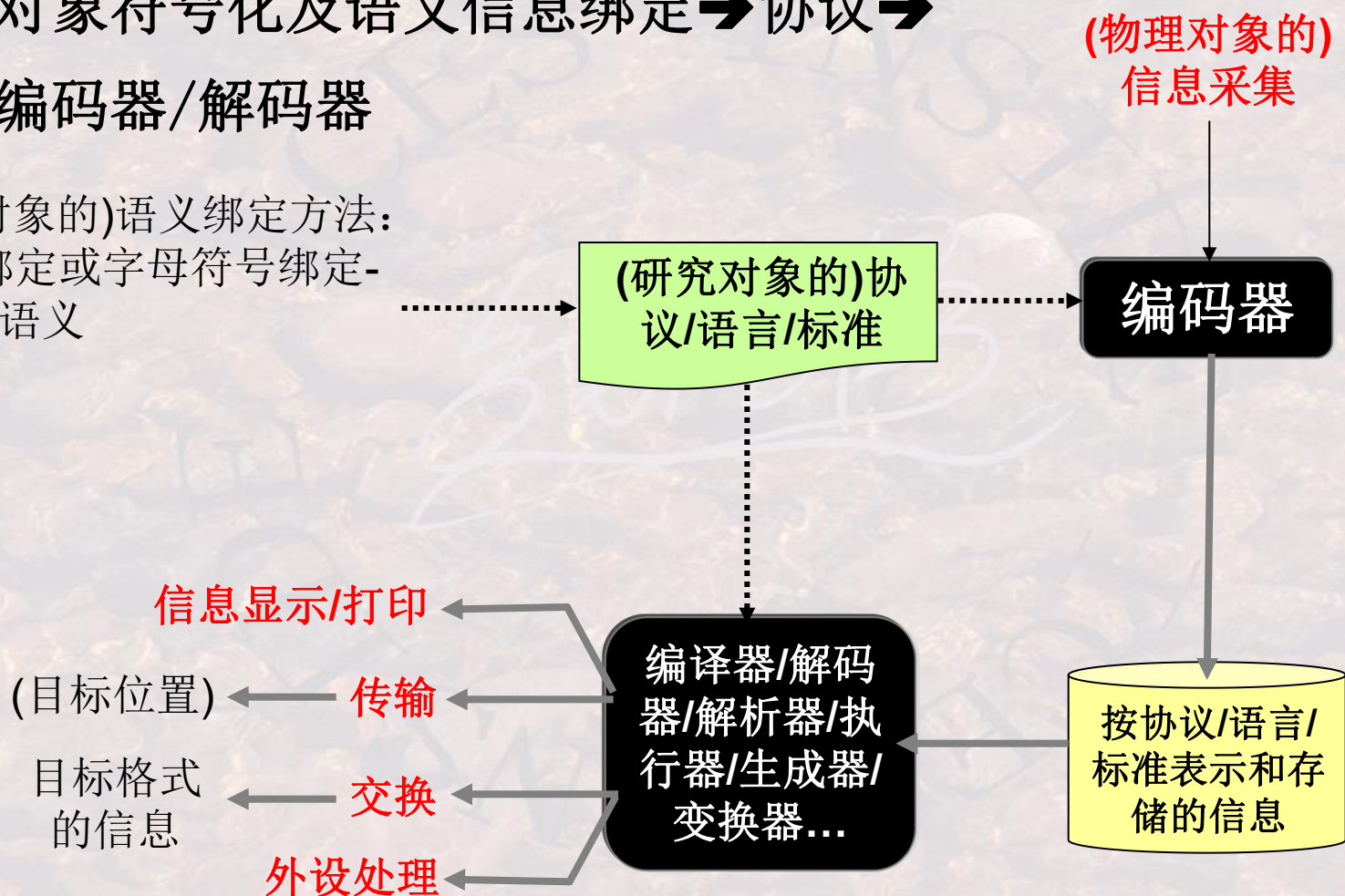
# 协议与编解码器--另一种抽象-自动化机制示例

## (3)推而广之

物理对象符号化及语义信息绑定→协议→

协议编码器/解码器

(物理对象的)语义绑定方法:  
0/1码绑定或字母符号绑定-  
---何种语义



协议与编解码器/解码器体现了信息处理的一般性思维



# 协议与编解码器--另一种抽象-自动化机制示例

## (4)多媒体处理是否也可采用这种思维呢？

### 协议与编解码器/解码器示例----多媒体处理

#### 音频信息的表示与处理

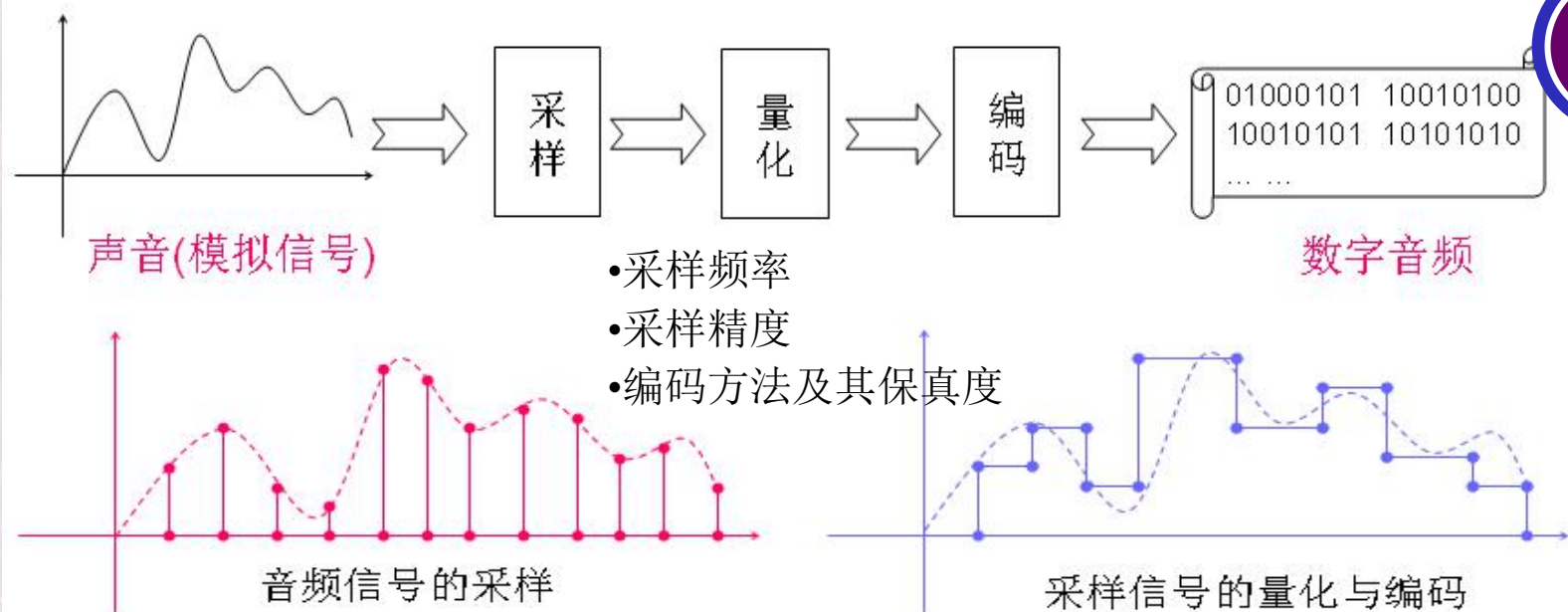
◆**音频编码**：音频是连续的模拟信号，需经采样、量化和编码后形成数字音频后，进行数字处理。所谓采样是指按一定的采样频率对连续音频信号做时间上的离散化，即对连续信号隔一定周期获取一个信号点的过程。而量化是将所采集的信号点的数值区分成不同位数的离散数值的过程。而编码则是将采集到的离散时间点的信号的离散数值按一定规则编码存储的过程。

采样

量化

编码

解码



### 协议与编解码器/解码器示例----多媒体处理

#### 图像的表示与处理

◆**位图图像**：将图像划分成均匀的由单元点构成的网格，每个单元点称为像素。每个像素可由1位或多位表示，1位只能表示黑白图像，8位能表示灰度图像，24位则能表示彩色图像。单位尺寸内的像素数目被称为图像的分辨率，由水平像素数目×垂直像素数目来表示。



像素表达颜色的不同，  
需要编码的位数不同。

黑白-1位(0,1)

256级灰度-8位(0-255)

16色彩色-4位(0-16)

256色彩色-8位(0-255)

24位真彩色-24位

(红0-255、绿0-255、蓝0-255三元色)

采样

量化

编码

解码

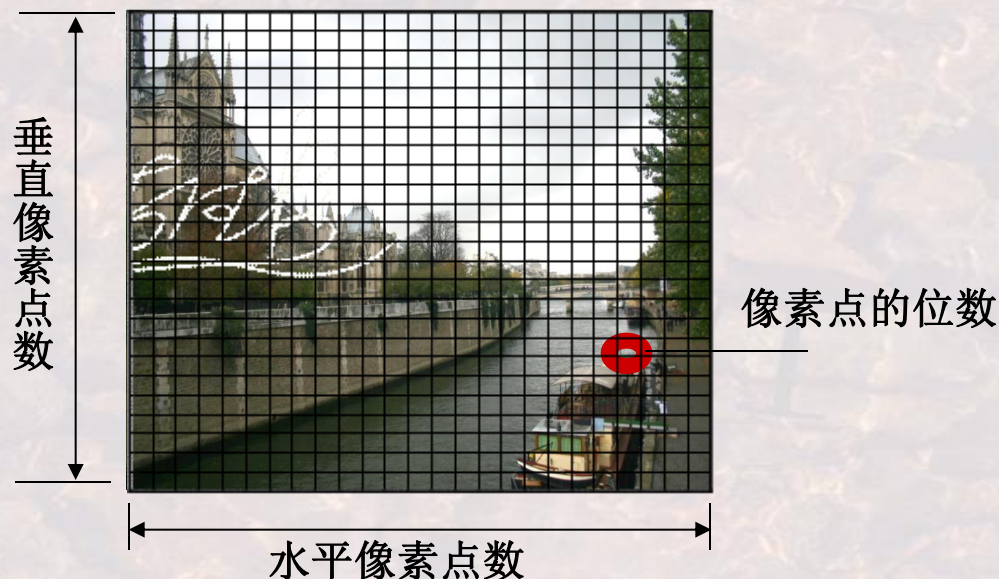


### 协议与编解码器/解码器示例----多媒体处理

◆ **图像编码：** 由于位图图像的存储量大(水平像素数目×垂直像素数目×每像素位数)，通常都需要进行压缩存储，不同的压缩采用了不同的图像编码。典型的有：

◆ **JPEG：** 国际标准化组织(ISO)和国际电报电话咨询委员会(CCITT)联合成立的“联合照片专家组”于1991年3月提出了JPEG标准(Joint Photographic Experts Group)。

◆ 其他常用编码格式有：**BMP，GIF，TIFF，...**



采样

量化

编码

解码

你知道图像的编码器、解码器是什么吗？你有使用过吗？



# 协议与编解码器--另一种抽象-自动化机制示例

## (4)多媒体处理是否也可采用这种思维呢？

### 协议与编解码器/解码器示例----多媒体处理

#### 视频信号的表示与处理

◆视频：是时间序列的动态图像(如25帧/秒)，也是连续的模拟信号，需要经过采样、量化和编码形成数字视频，保存和处理。同时，视频还可能由视频、音频及文字经同步后形成的。因此视频处理相当于按照时间序列处理图像、声音和文字及其同步问题。

采样

量化

编码

解码



# 协议与编解码器--另一种抽象-自动化机制示例

## (4)多媒体处理是否也可采用这种思维呢？

### 协议与编解码器/解码器示例----多媒体处理

#### 视频信号的表示与处理

◆视频编码：MPEG是Moving Pictures Experts Group(动态图象专家组)的缩写。提出了四个版本：MPEG-I(VCD: Video CD)、MPEG-II (DVD:Digital Versatile Disk)、MPEG-III、MPEG-IV(多媒体)。

◆MIDI 音乐(Musical Instrument digital Interface), WAV, MP3等是音频的编码标准。

采样

量化

编码

解码



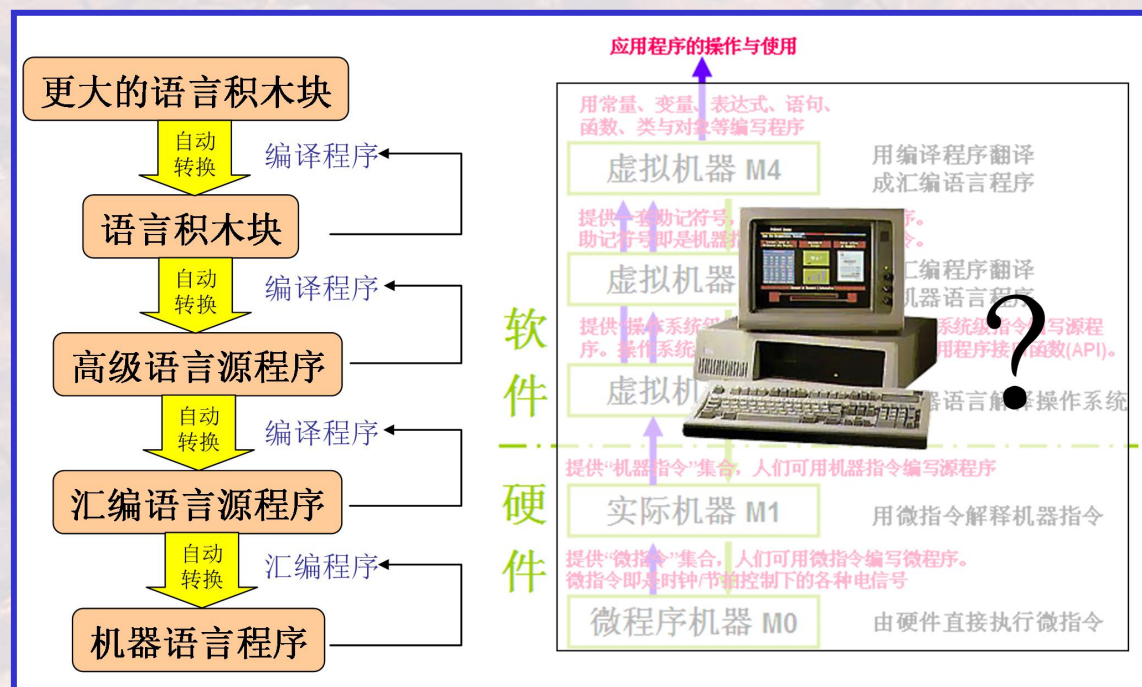


# 协议与编解码器--另一种抽象-自动化机制示例

## (7)小结

基本目标: 理解如何编写计算机可以执行的程序

执行零壹编程符号,  
汇编语言高级语言。  
不同层次虚拟机器,  
分层变换编译真谛。  
各级协议编解码器,  
道理相通扩展容易。



基本思维: 高级语言与汇编语言 → 语言与编译器 → 协议与编解码器/解法器 → 不同层面的计算机;  
相关课程: 《高级语言程序设计》 《汇编语言程序设计》 《编译原理》 《计算机网络》 《操作系统》 《Internet》;