

## 第2章 早期人工智能技术

苏统华  
软件学院  
哈尔滨工业大学

## 本章内容

1

传统人工智能GOFAI

2

专家系统

## 2.1 传统人工智能GOFAI

**苏统华**  
软件学院  
哈尔滨工业大学

# 符号主义概述

## □ 19世纪中期

- ✓ **乔治·布尔 (George Boole)**: 通过符号操作实现逻辑推理, 发展了布尔代数。
- ✓ **哥特洛布·弗雷格 (Gottlob Frege)**: 提出了谓词逻辑, 将逻辑推理形式化为符号系统。

## □ 20世纪初

- ✓ **伯特兰·罗素 (Bertrand Russell)** 和 **阿尔弗雷德·怀特海 (Alfred North Whitehead)**: 合作撰写《数学原理 (Principia Mathematica)》, 推动符号逻辑的发展。

## □ 20世纪30年代

- ✓ **艾伦·图灵 (Alan Turing)**: 提出图灵机 (Turing Machine) 概念, 证明了任何计算过程都可以通过符号操作实现, 为计算科学奠定理论基础。

## □ 20世纪50年代至80年代末

- ✓ **符号主义 (Symbolic AI)**: 在计算机科学和人工智能领域中占据主导地位, 成为人工智能研究的三大主流派系之一。

## 艾伦·纽厄尔、赫伯特·西蒙与物理符号系统

### □物理符号系统假设(PSSH)

- ✓ 物理符号系统具备实现通用智能的必要且充分条件。
- ✓ 人类思维本质是一种符号处理过程。
- ✓ 符号系统足以实现智能，特定功能在计算机上实现即可满足。

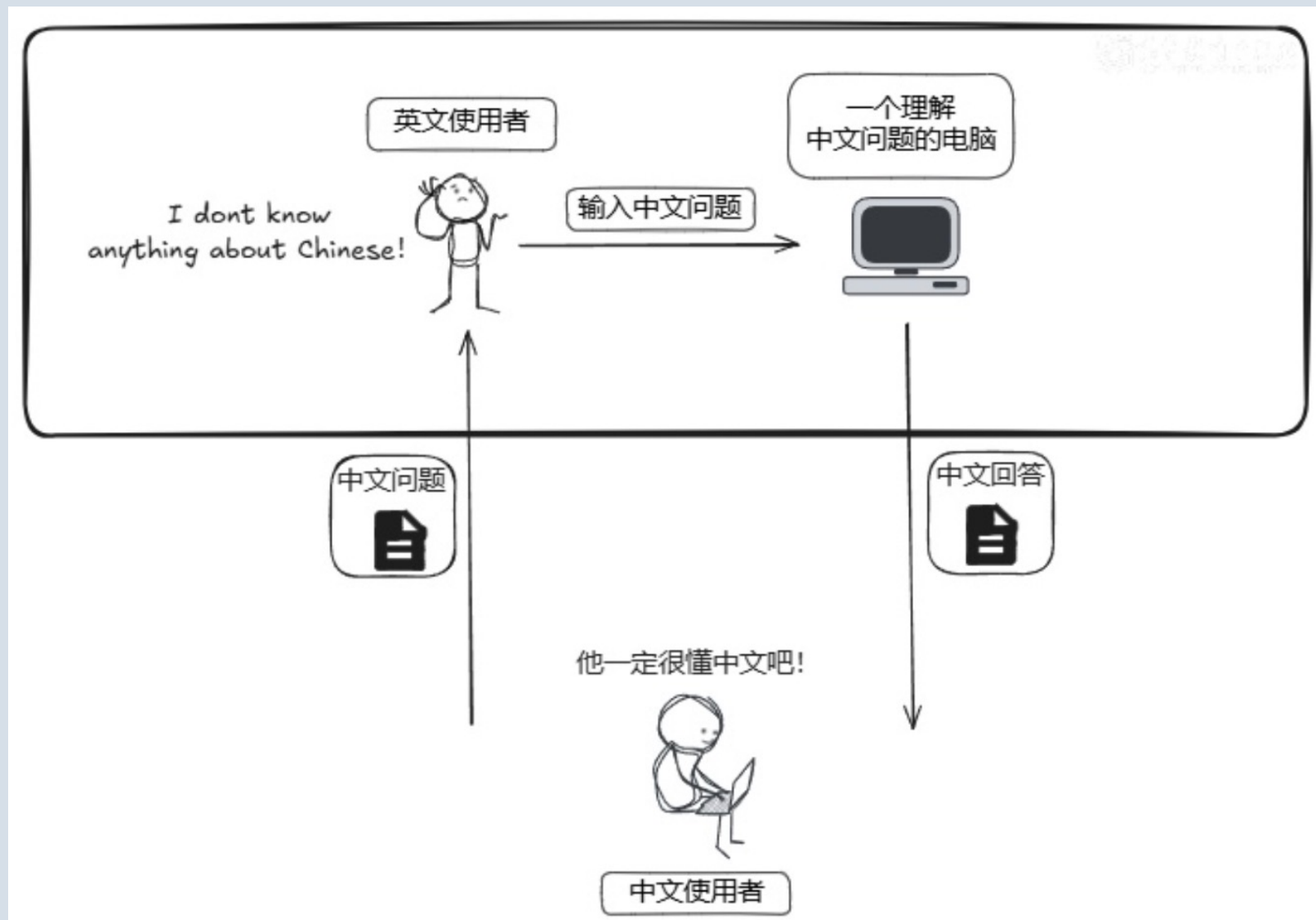
### □物理符号系统的例子

- ✓ **形式逻辑**：包括“与”“或”“非”等逻辑词，处理程序遵循逻辑推论规则。
- ✓ **代数**：包括“+”“×”等符号，表示数字关系，操作遵循代数规则。
- ✓ **数字计算机**：符号是存储的指令和数据，处理程序是指令的执行。
- ✓ **国际象棋**：符号是棋子和棋盘，程序是棋子移动规则。



## 中文屋实验

- 美国哲学家约翰·塞尔 (John Searle) 1980年设计
- 在这个实验中，塞尔设想自己被关在一个房间里，通过一本中文指令手册，根据输入的中文字符序列，生成相应的中文输出。尽管塞尔不懂中文，但他依然能够通过指令手册正确地处理输入并生成合理的输出。
- 塞尔的核心论点是，即使他能够完美地执行这些指令，他仍然不理解中文，只是在机械地操作符号
- 同样地，塞尔认为计算机程序也只是处理符号，并不具备真正的理解或意识。



## 知识与知识表达：IF-THEN规则表达式

### □ IF（条件）：

- ✓ 指定在何种条件下规则应被激活。条件部分通常描述了一系列触发规则的前提。

### □ THEN（结论）：

- ✓ 定义在条件满足时需要采取的行动或产生的结果。结论部分描述了应对特定条件的具体操作或输出。

IF      <条件>      THEN      <结论>

## 知识与知识表达：IF-THEN规则表达式

□ **IF** 患者有咳嗽 AND 发烧 **THEN** 患者可能有流感。

- ✓ 这个规则表示，当患者同时出现咳嗽和发烧两个症状时，有可能患上了流感。这种简单的推理可以帮助医生快速初步诊断，虽然可能需要进一步的检查来确认。

□ **IF** 作物叶子发黄 AND 土壤湿度低 **THEN** 需要给作物浇水。

- ✓ 这个规则表明，当作物叶子发黄且土壤湿度低时，通常是因为缺水，因此需要及时浇水。这有助于农民及时采取措施，避免作物损失。

□ **IF** 月收入大于月支出 **THEN** 进行储蓄。

- ✓ 该规则表示，如果一个人的月收入大于月支出，那么建议进行储蓄。这可以帮助个人或家庭更好地管理财务，确保资金合理地使用和积累。



## 知识与知识表达：正向推理

•R1: IF A and B THEN C

•R2: IF C and D THEN E

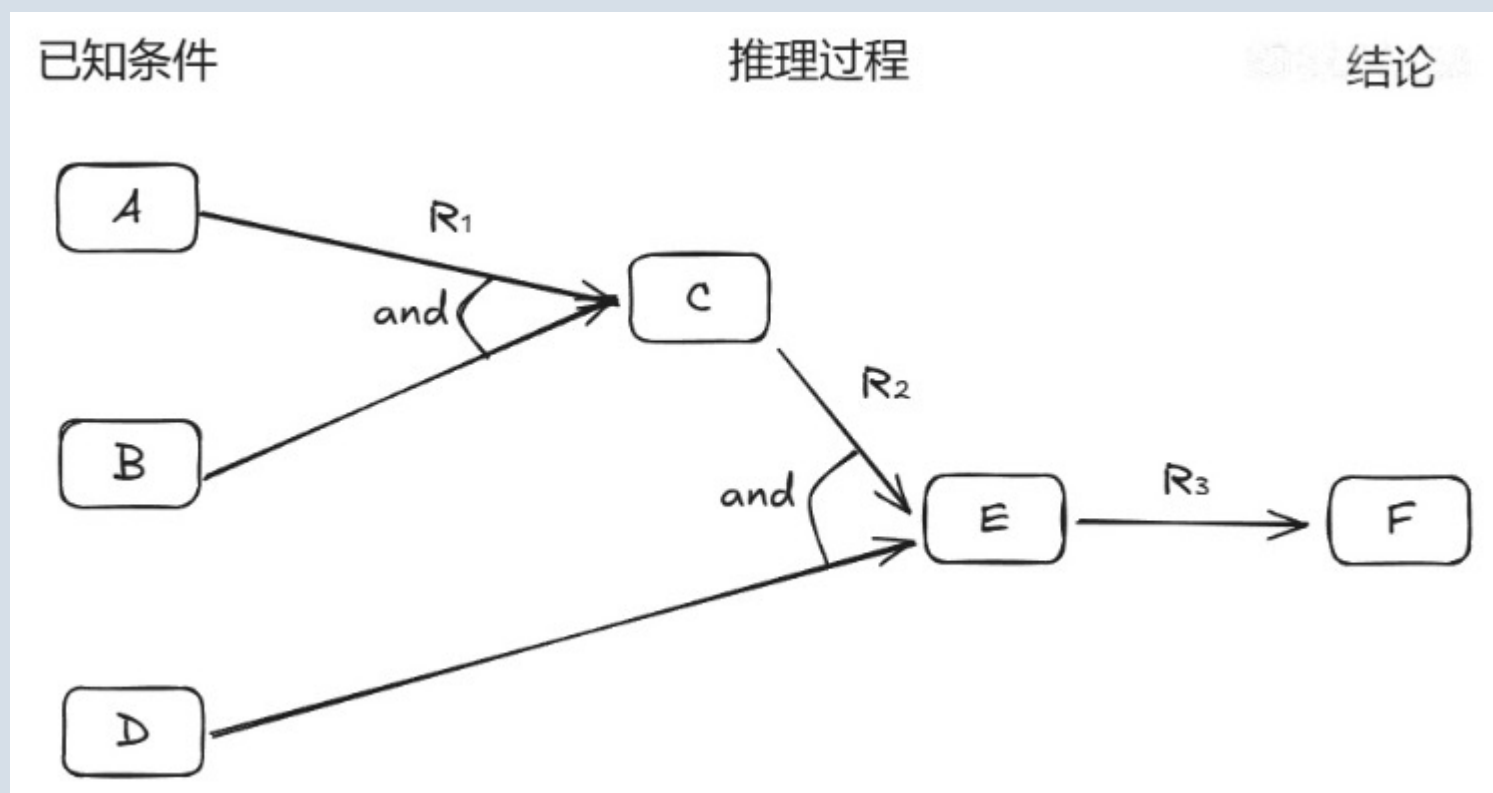
•R3: IF E THEN F

并且已知A、B、D已经成立，求证F成立。

·初始已知A、B成立，可以根据R1推出C成立；

·初始已知D成立，且已经推出C成立，则可以推出E成立；

·现已推出E成立，则可以根据R3推出F成立



## 知识与知识表达：正向推理

•R1: IF A and B THEN C

•R2: IF C and D THEN E

•R3: IF E THEN F

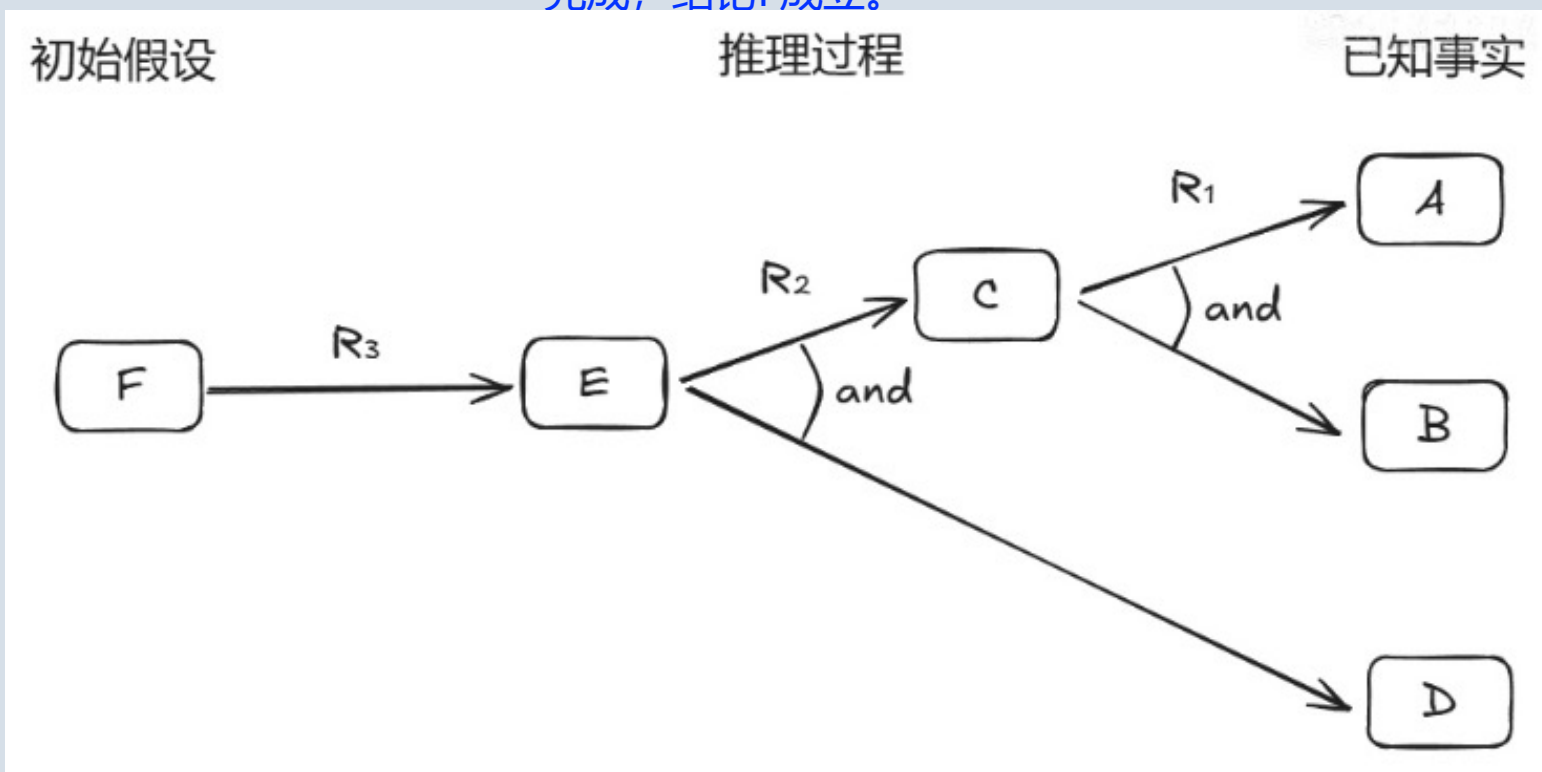
并且已知A、B、D已经成立，求证F成立。

·首先假设F成立，对已有规则进行匹配，发现R3的结论为F，则可以假设E成立；

针对结论为E的规则进行匹配，发现R2的结论为E，则可以假设C和D成立；

·由于已知D是成立的，因此只需要继续考虑C是否成立，针对结论为C的规则进行匹配，发现R1的结论为C，则假设A和B成立；

·由于已知A和B是成立的，目前所有的假设都已经推理出是已知成立的，则推理完成，结论F成立。



# ELIZA: 早期自然语言处理程序

## □ ELIZA的背景

- ✓ ELIZA 是最早的自然语言处理程序，由系统工程师维吉尔·弗里德曼设计。
- ✓ 它的目标是模拟人类的对话方式，尤其通过使用规则和模式模拟心理治疗的回话。

## □ 设计理念

- ✓ 通过使用GOFAI系统设计对话规则。
- ✓ ELIZA通过定义一系列规则来处理输入，选择关键字并通过规则生成回应。

## □ 运作方式

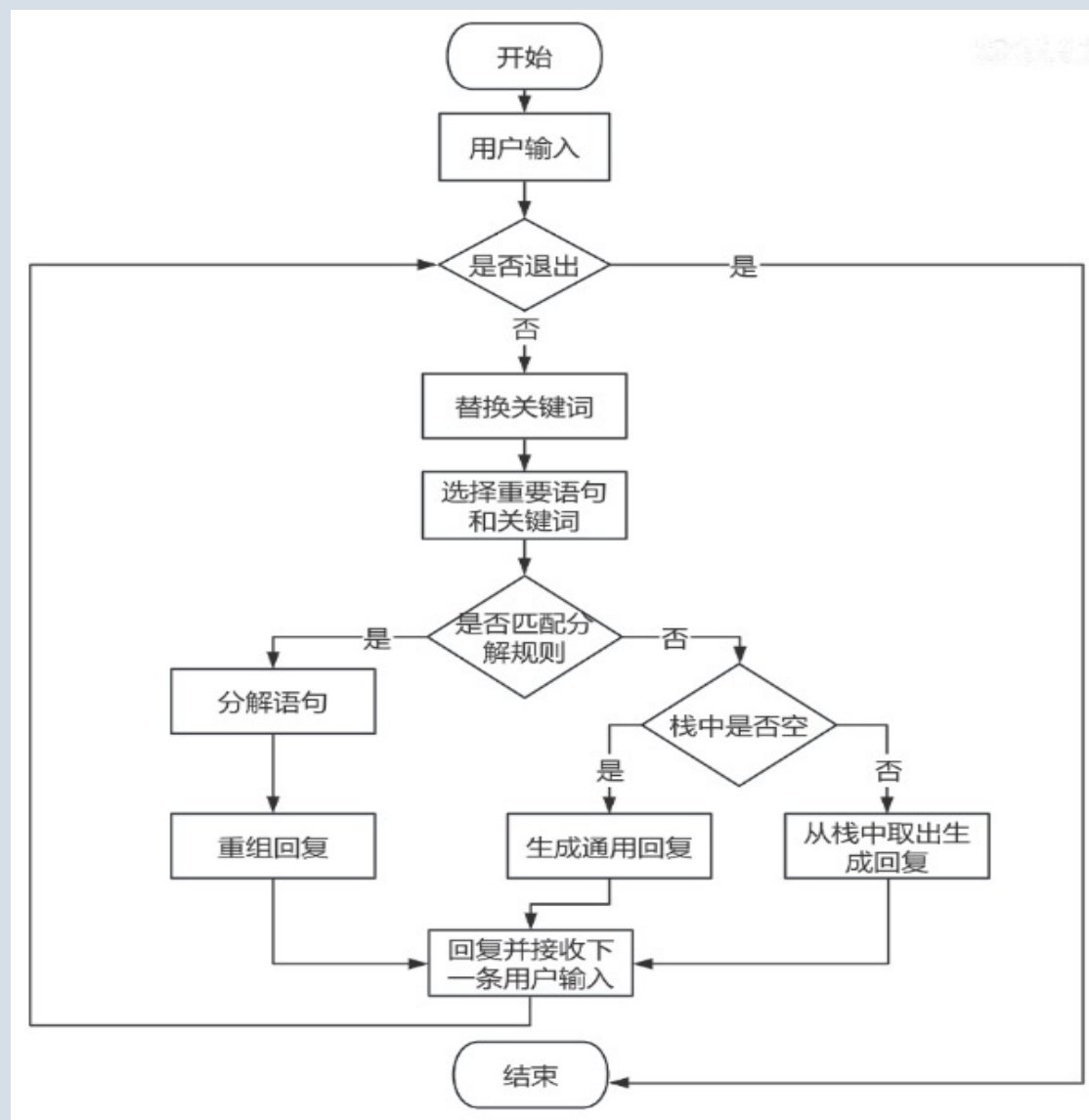
- ✓ 使用“假设”规则：系统通过预设规则对输入进行模式匹配，根据这些规则生成回应。

## □ 技术特点

- ✓ 使用MAD (Michigan Algorithm Decoder) 和SLIP (Symmetric List Processor) 编程语言
- ✓ 这些编程语言在20世纪60年代用于开发ELIZA，并为其处理自然语言提供支持。

	PRINT, TQ109, 2531, SPEAK, MAD	TQ109 2531
	CHANGE MAD	
	EXTERNAL FUNCTION (KEY, MYTRAN)	000010
	NORMAL MODE IS INTEGER	000020
	ENTRY TO CHANGE.	000030
	LIST.(INPUT)	000040
	V'S G(1)=\$TYPE\$, \$SUBST\$, \$APPEND\$, \$ADD\$,	000050
	1\$START\$, \$RANK\$, \$DISPLA\$	000060
	V'S SNUMB = \$ I3 *\$	000070
	FII=0	000080
CHANGE	PRINT COMMENT \$PLEASE INSTRUCT ME\$	001400
	LISTRD.(MYLIST.(INPUT), 0)	001410
	JOB=POPTOP.(INPUT)	001420
	T'H IDENT, FOR J=1, 1, J.G. 7	001430
IDENT	W'R G(J) .E. JOB, T'O THEMA	001440
	PRINT COMMENT \$CHANGE NOT RECOGNIZED\$	001450
	T'O CHANGE	001460
THEMA	W'R J .E. 5, F'N IRALST.(INPUT)	001470
	W'R J .E. 7	001480
	T'H DISPLA, FOR I=0, 1, I .G. 32	001490
	W'R LISTMT.(KEY(I)) .E. 0, T'O DISPLA	001500
	S=SEQRDR.(KEY(I))	001510
READ(7)	NEXT=SEQLR.(S, F)	001520
	W'R F .G. 0, T'O DISPLA	001530
	PRINT COMMENT **\$	001540
	TPRINT.(NEXT, 0)	001550
	PRINT FORMAT SNUMB, I	001560
	PRINT COMMENT \$ \$	001570
	T'O READ(7)	001580
DISPLA	CONTINUE	001590
	PRINT COMMENT \$ \$	001600
	PRINT COMMENT \$MEMORY LIST FOLLOWS\$	001610
	PRINT COMMENT \$ \$	001620
	T'H MEMLST, FOR I=1, 1, I .G. 4	001630
MEMLST	TXTPRT.(MYTRAN(I), 0)	001640
	T'O CHANGE	001650
	E*L	001660
	THEME=PCPTOP.(INPUT)	001670
	SUBJCT=KEY(HASH.(THEME, 5))	001680
	S=SEQRDR.(SUBJCT)	001690
LOOK	TERM=SEQLR.(S, F)	001700
	W'R F .G. 0, T'O FAIL	001710
	W'R TOP.(TERM) .E. THEME, T'O FOUND	001720
	T'O LOOK	001730
FOUND	T'O DELTA(J)	001740
DELTA(1)	TPRINT.(TERM, 0)	001750
	T'O CHANGE	001760
FAIL	PRINT COMMENT \$LIST NOT FOUND\$	001770
	T'O CHANGE	001780
DELTA(2)	S=SEQRDR.(TERM)	001790
	OLD=POPTOP.(INPUT)	001800
READ(1)	OBJCT=SEQLR.(S, F)	001810
	W'R F .G. 0, T'O FAIL	001820
	W'R F .NE. 0, T'O READ(1)	001830
	INSIDE=SEQRDR.(OBJCT)	001840
READ(2)	IT=SEQLR.(INSIDE, F)	001850
	W'R F .G. 0, T'O READ(1)	001860
	SIT=SEQRDR.(IT)	001870
	SOLD=SEQRDR.(OLD)	001880
ITOLD	TOLD=SEQLR.(SOLD, FOLD)	001890

# ELIZA工作流程



## ELIZA对话实例

Eliza: Welcome.  
You: Men are all alike.  
Eliza: In what way?  
You: They're always bugging us about something or other.  
Eliza: Can you think of a specific example?  
You: Well, my boyfriend made me come here.  
Eliza: Your boyfriend made you come here?  
You: He says I'm depressed much of the time.  
Eliza: I am sorry to hear that you are depressed.  
You: It's true. I am unhappy.  
Eliza: Do you think coming here will help you not to be unhappy?  
You: I need some help, that much seems certain.  
Eliza: What would it mean to you if you got some help?  
You: Perhaps I could learn to get along with my mother.  
Eliza: Tell me more about your family.  
You: My mother takes care of me.  
Eliza: Who else in your family takes care of you?  
You: My Father.  
Eliza: Your Father?  
You: You are like my father in some ways.  
Eliza: What resemblance do you see?  
You: You are not very aggressive but I think you don't want me to notice that.  
Eliza: What makes you think I am not very aggressive?  
You: You don't argue with me.  
Eliza: Why do you think I don't argue with you?  
You: You are afraid of me.  
Eliza: Does it please you to believe I am afraid of you?  
You: My father is afraid of everybody.  
Eliza: What else comes to mind when you think of your father?  
You: Bullies.  
Eliza: Earlier you mentioned your father.

# 传统人工智能的局限性

## □ 组合爆炸问题

- ✓ 组合爆炸指的是当元素数量有限但不断增加时，所形成的组合会极速增长的现象
- ✓ 传统人工智能依赖符号系统和逻辑运算，当需要处理的数据增长，会导致运算资源需求不断增加

## □ 机器翻译困难

- ✓ 传统人工智能在机器翻译上，本质是运用符号逻辑的一种算法。
- ✓ 在实际运用中，由于语言的地区性和使用习俗特点，会导致歧义
- ✓ 无法正确理解上下文

## □ 常识匮乏

- ✓ 常识是指日常普遍使用，广泛有效且无需证明的知识
- ✓ 常识具有与语境密切相关的特征，而传统人工智能无法识别这些特征
- ✓ 常识数据规模的庞大超出了传统人工智能模型处理的能力

## 2.2 专家系统

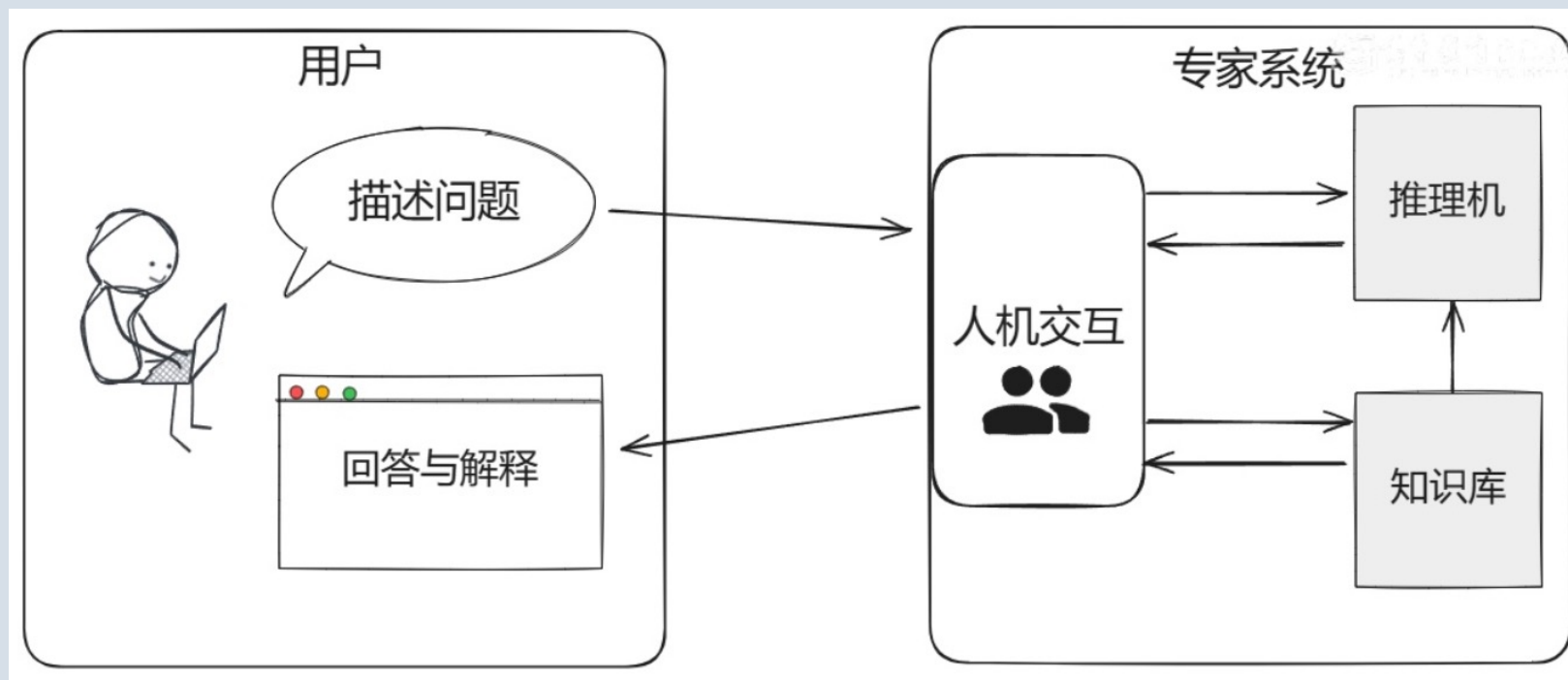
**苏统华**  
软件学院  
哈尔滨工业大学



## 专家系统抽象架构

### □ 专家系统抽象架构

- ✓ 用户通过人机交互界面向系统提出问题描述。
- ✓ 系统将问题输入推理机，推理机利用知识库中的已有知识进行逻辑推理。
- ✓ 推理结果（包括回答和解释）返回给人机交互界面并展示给用户。
- ✓ 知识库内容需要定期由人类专家进行更新，确保系统与知识库保持连接。





## 专家系统基本结构

### □ 知识库：

- ✓ 核心组件，存储专业知识。
- ✓ 使用IF-THEN规则表示信息。
- ✓ 决定系统的专业性和适用范围。

### □ 推理机：

- ✓ 系统的引擎，进行逻辑推理和决策。
- ✓ 处理明确规则和不确定性问题。

### □ 动态数据库：

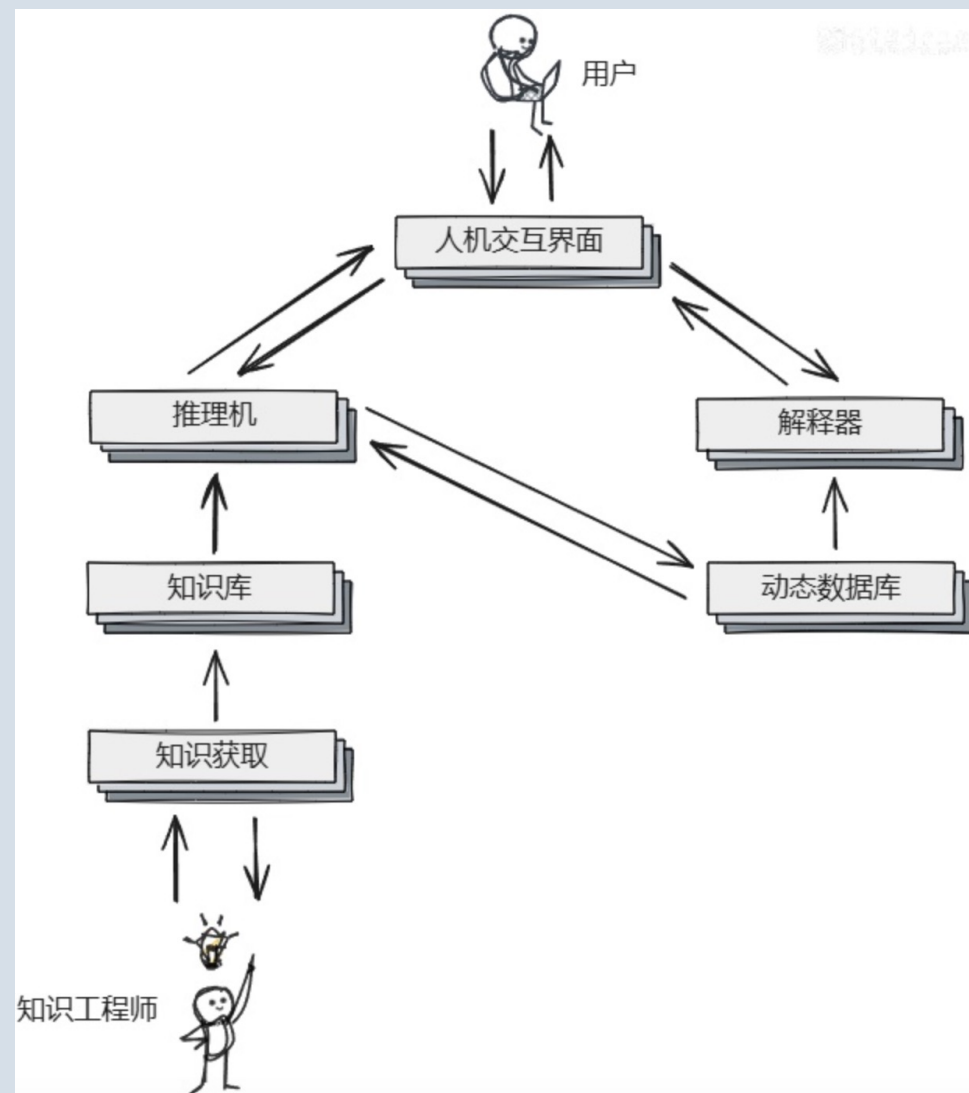
- ✓ 存储推理过程中的临时数据和中间结果。
- ✓ 提升系统的灵活性和响应速度。

### □ 解释器：

- ✓ 解释推理过程和决策结果。
- ✓ 提供“Why解释”和“How解释”，增强透明度。

### □ 人机交互界面：

- ✓ 用户与系统之间的沟通桥梁。
- ✓ 接收用户输入并反馈推理结果。



## 专家系统结果获取流程

### 1.用户输入:

- 用户通过人机交互接口输入问题。

### 2.问题转换:

- 人机交互界面将问题转换为知识库的表达形式,并输入推理机。

### 3.推理过程:

- 推理机进行推理,得出结果。

### 4.结果存储:

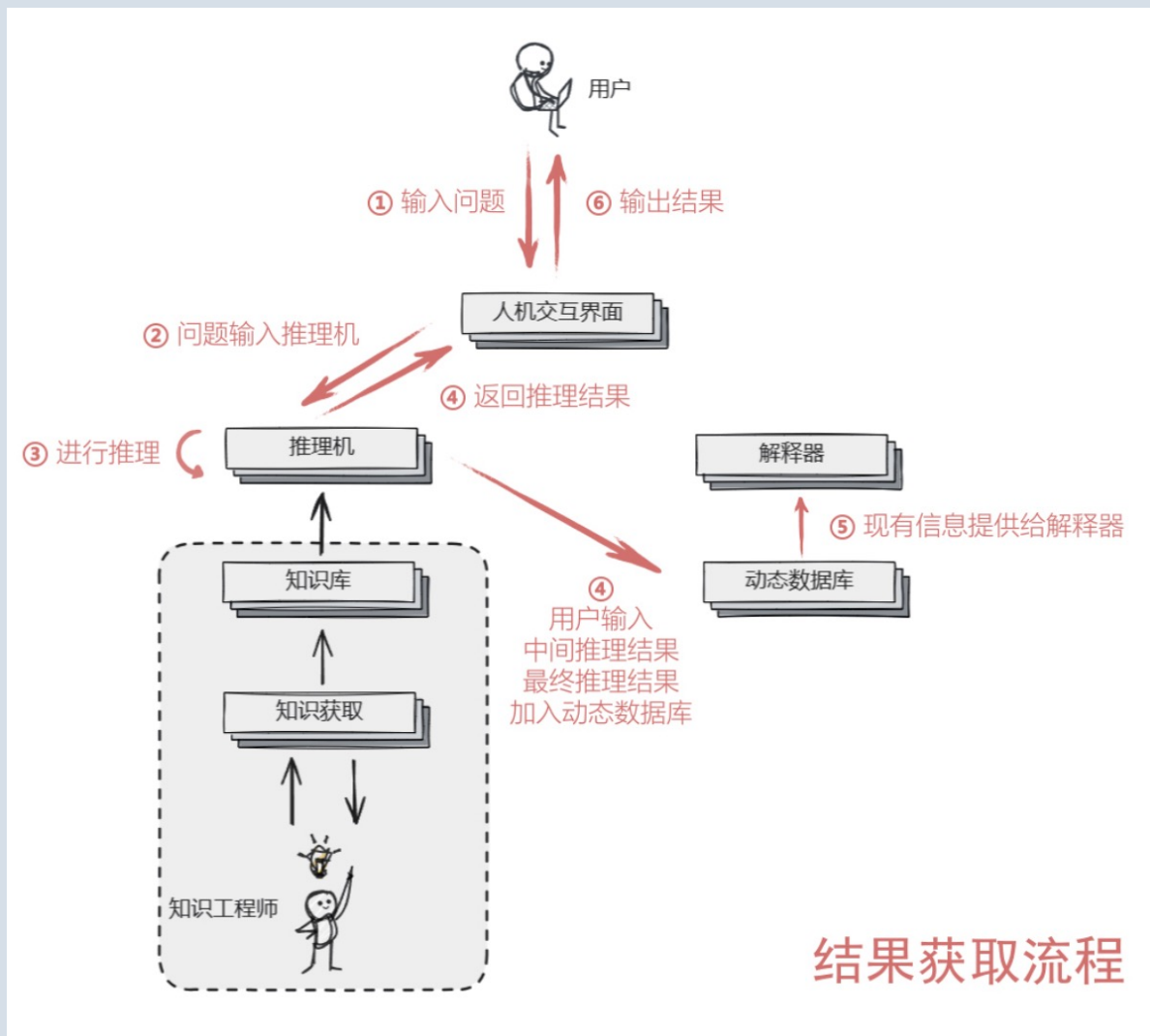
- 推理机将最终结果返回给人机交互界面,并将相关信息存入动态数据库。

### 5.解释准备:

- 动态数据库将信息传递给解释器,以备用户请求解释时使用。

### 6.结果输出:

- 人机交互界面输出推理结果,并告知用户。



结果获取流程

## 专家系统结果获取流程

### 1.用户请求:

- 用户通过人机交互接口请求解释。

### 2.解释申请:

- 人机交互界面向解释器申请生成解释。

### 3.解释生成:

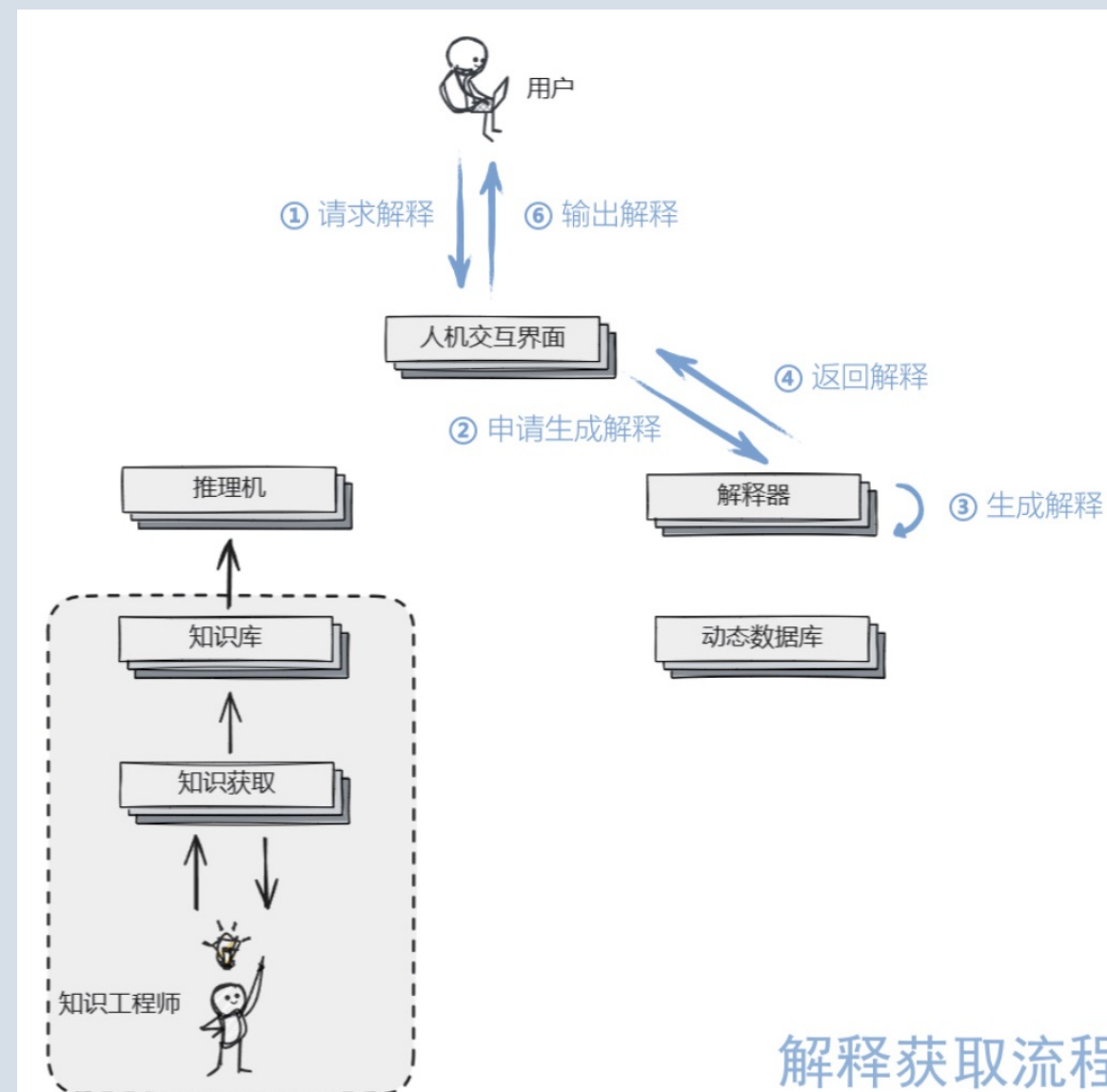
- 解释器根据已有信息生成对推理结果的解释。

### 4.解释返回:

- 解释器将生成的解释信息返回给人机交互界面。

### 5.解释输出:

- 人机交互界面向用户输出解释信息。



# MYCIN---第一个成熟的专家系统

## 1.MYCIN系统：

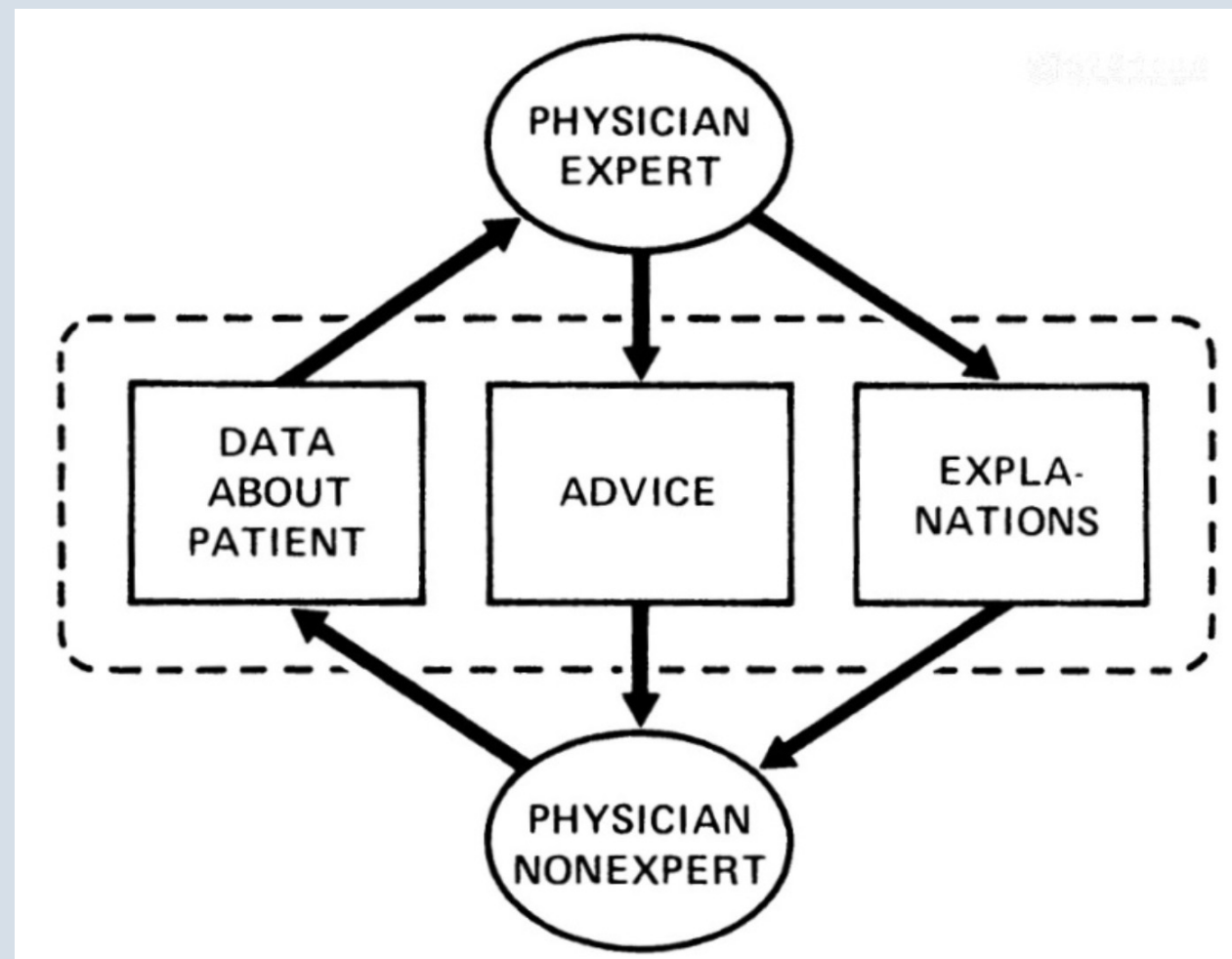
- 历史上第一个成熟的专家系统，用于诊断严重感染症状并推荐抗生素使用。
- 由艾德华·H·肖特利夫等人于1972年研发，误诊率低于当时的人类医生。
- 肖特利夫在1975年发表了相关文章，并在1976年出版了详细书籍。

## 2.系统设计：

- 分为抽象架构和基本结构。
- 基于规则知识，专门用于医疗诊断。

## 3.信息传递与交互：

- 系统在医学专家和非专家医生之间传递信息。
- 核心模块包括患者数据、建议和解释。
- 系统接收患者数据，生成诊断建议，并通过解释模块增强透明性和可信度。



# MYCIN基本结构

## 1. MYCIN系统内部结构：

- 划分为静态知识（知识库）、动态知识（动态数据库）和规则解释器（推理机）三个模块。
- 这些模块为后续专家系统的模块化设计奠定了基础。

## 2. 工作原理：

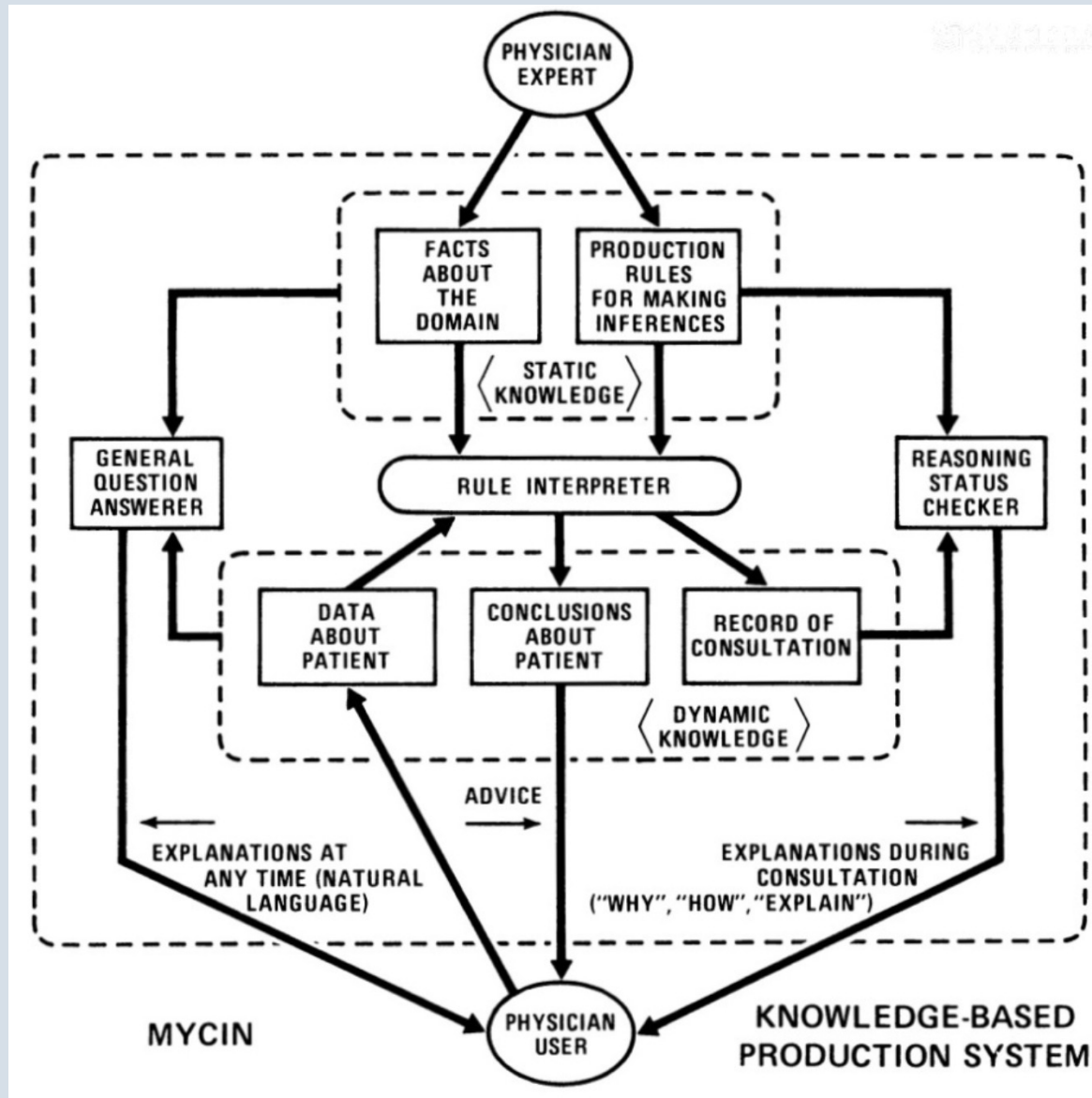
- 医学专家提供领域事实和生成规则，归类为静态知识。
- 规则解释器基于静态知识分析患者数据，生成患者结论。
- 推理状态检查器实时监控推理过程，确保合理性。
- 系统记录整个咨询过程，归类为动态知识，供后续参考。

## 3. 解释模块：

- 包含常规问题回答模块，随时提供自然语言解释。
- 咨询过程中提供详细解释，回答“Why”和“How”等问题。

## 4. 诊断建议：

- 系统生成的诊断建议传递给使用者，帮助作出治疗决策。



# 非确定性事件表示

## 1.非确定性事件表示：

- 事件 A 的置信因子 (CF) 表示为  $CF(A)$ ，取值范围为  $[-1,1]$ 。
- $CF(A)=1$ ：事件 A 一定成立。
- $CF(A)\in(0,1)$ ：事件 A 以一定置信度为真。
- $CF(A)=0$ ：系统对事件 A 的真假无信息。
- $CF(A)\in[-0.2,0.2]$ ：通常认为对事件 A 一无所知。
- $CF(A)\in(-1,0)$ ：事件 A 为假的置信度为  $-CF(A)$ 。
- $CF(A)=-1$ ：事件 A 一定为假。

## 2.示例：

- $CF(\text{下雪}) = 0.8$ ：下雪的置信度为 0.8。
- $CF(\text{下雪}) = -0.8$ ：不下雪的置信度为 0.8。

# 非确定性知识表示

## 1.非确定性知识表示：

- 形式：IF A THEN B CF(B,A)
- A：条件；B：结论；CF(B,A)：置信度（知识的强度）。
- CF(B,A) 的取值范围和含义：
  - ✓  $CF(B,A) \in (0,1]$ ：条件与结论正相关。
  - ✓  $CF(B,A)=0$ ：系统对该条知识一无所知，条件与结论无联系。
  - ✓  $CF(B,A) \in [-1,0]$ ：条件与结论负相关。

## 2.示例：

1. IF 冬天 THEN 下雪 0.8：冬天时下雪的概率为0.8。
2. IF 夏天 THEN 下雪 -0.8：夏天时下雪的概率为-0.8，即不下雪的概率为0.8。
3. IF 上学 THEN 下雪 0：上学和下雪之间无联系。

## 3.复合条件：

IF 发烧 and 咳嗽 and 流鼻涕 THEN 流感 0.7：  
同时有发烧、咳嗽和流鼻涕时，确诊流感的概率为0.7。



# 条件逻辑运算

## 1.条件的逻辑运算：

复合条件可以通过逻辑运算表示，常用运算包括“与”（and）、“或”（or）和“非”（not）。

## 2.逻辑运算规则：

- **与（and）**：  $CF(A \text{ and } B) = \min CF(A), CF(B)$ ，表示事件 A 与 B 同时成立的置信度，取  $CF(A)$  和  $CF(B)$  中最小的一个。
- **或（or）**：  $CF(A \text{ or } B) = \max CF(A), CF(B)$ ，表示事件 A 或 B 成立的置信度，取  $CF(A)$  和  $CF(B)$  中最大的一个。
- **非（not）**：  $CF(\text{not } A) = -CF(A)$ ，表示事件 A 不成立的置信度，取事件 A 置信度的负值。



## 条件不为真的非确定性知识表示

### 1.条件不为真的非确定性知识表示：

- 规则知识的置信度基于条件为真。
- 若条件  $CF(A)$  不为1，需确定结论  $CF(B)$ 。

### 2.结论置信度的计算方法：

- 已知：IF A THEN B  $CF(B,A)$
- 公式： $CF(B)=\max[0,CF(A)]\times CF(B,A)$
- 只有当  $CF(A) > 0$  时，才能推出结论  $CF(B)$ 。

### 3.重要前提：

- 使用  $\max\{0,CF(A)\}$  确保  $CF(A)$  必须大于0。
- 若  $CF(A) < 0$ ，则  $CF(B) = 0$ ，表示该规则知识无法提供与 B 相关的信息。

### 4.注意事项：

$CF(B)=0$  并不一定表示系统对 B 一无所知，可能有其他规则知识推导出与 B 相关的信息。

# 规则知识合成与非确定性推理

## 1. 规则知识合成与非确定性推理：

- 推理机的核心是规则知识合成与非确定性推理。
- 非确定性推理过程即 CF 系统下的规则知识合成过程。

## 2. 多条规则知识推导同一结论：

- 存在多种前提条件推导出同一结论的情况。
- 示例：
  - 第1条规则知识: IF 发烧 THEN 流感 0.8
  - 第2条规则知识: IF 流鼻涕 THEN 流感 0.5
  - $CF(\text{发烧}) = 0.5$
  - $CF(\text{流鼻涕}) = 0.4$

## 3. 结论置信度的计算：

3. 由第1条规则知识推出  $CF(\text{流感}) = 0.8 \times 0.5 = 0.4$

4. 由第2条规则知识推出  $CF(\text{流感}) = 0.5 \times 0.4 = 0.2$

## 4. 规则知识合成方法：

- 设  $CF_1(B)$  和  $CF_2(B)$  分别为两条规则知识推出的结论 B 的置信度因子。
- 合成公式：
$$CF(B) = CF_1(B) + CF_2(B) \times (1 - |CF_1(B)|)$$
- 示例合成结果： $CF(\text{流感}) = 0.4 + 0.2 \times (1 - 0.4) = 0.52$

## 5. 多个规则知识的合成：

采用类似二叉树的合成方法，逐步合成多个规则知识。

## Mini-MYCIN规则知识

- ① IF 温度高 and 咳嗽 THEN 流感 0.8
- ② IF 温度高 and 头痛 THEN 流感 0.6
- ③ IF 咳嗽 and 喉咙痛 THEN 感冒 0.7
- ④ IF 头痛 and 恶心 THEN 偏头痛 0.9
- ⑤ IF 皮疹 and 发痒 THEN 过敏 0.7
- ⑥ IF 发烧 and 疲劳 THEN 病毒感染 0.75
- ⑦ IF 肌肉疼痛 and 疲劳 THEN 流感 0.7
- ⑧ IF 流鼻涕 and 咳嗽 THEN 感冒 0.65
- ⑨ IF 咳嗽 and 恶心 THEN 胃炎 0.6
- ⑩ IF 疲劳 and 头痛 THEN 偏头痛 0.8
- ⑪ IF 温度高 and 咳嗽 and 肌肉疼痛 THEN 流感 0.9
- ⑫ IF 发烧 and 头痛 and 疲劳 THEN 病毒感染 0.8

## Mini-MYCIN推理机工作方式---询问驱动式的逆向推理

### 1. 假设生成与验证

- 系统基于假设规则启动推理，生成初始假设（如患者患“流感”）。
- 按存储顺序检索相关知识并开始验证。

### 2. 症状查询与动态数据存储

- 询问患者相关症状（如“温度高”、肌肉疼痛等）。
- 根据回答动态存储信息，逐步排除不符合条件的知识。

### 3. 逐步诊断与条件验证

- 若当前假设不成立，切换至下一规则继续验证（如“感冒”）。
- 动态评估患者症状，判断是否满足特定知识条件。

### 4. 综合判断与结论输出

- 动态知识库汇总“已知事实”，结合假设的CF因子（置信度）得出诊断结论。
- 系统提供最终的假设准确性及原因分析。

## Mini-MYCIN推理机工作方式---输入匹配式的正向推理

### 1. 信息驱动诊断

- 用户需一次性输入所有已知信息（如“温度高”、“咳嗽”、“头痛”）。
- 系统通过正向匹配规则自动得出诊断结论。

### 2. 知识库匹配

- 匹配输入症状与知识库规则：
- “温度高”+“咳嗽” → 推导“流感”（知识①）。
- “温度高”+“头痛” → 也推导“流感”（知识②）。

### 3. 规则合并与置信度计算

- 将符合条件的知识合并。
- 输出综合诊断结论及置信度（CF因子）。

### 4. 直接输出结果

- 系统基于匹配的知识规则直接推断结果并告知用户诊断的准确率。

# Mini-MYCIN解释器知识库

诊断	推荐用药	推荐用药原因解释
流感	奥司他韦	奥司他韦中的活性成分神经氨酸酶抑制剂能够抑制病毒在体内的复制，从而有效减轻流感症状
感冒	复方氨酚烷胺片	复方氨酚烷胺片是针对普通感冒的常用药物，能够缓解鼻塞、咳嗽、喉咙痛等感冒症状
偏头痛	舒马普坦	舒马普坦是一种选择性5-HT1受体激动剂，可以收缩颅内血管，缓解偏头痛
过敏	氯雷他定	氯雷他定是一种非镇静性抗组胺药，可以有效缓解过敏症状，如皮疹和瘙痒
病毒感染	对乙酰氨基酚	对乙酰氨基酚通过抑制前列腺素的合成来减轻发烧和疲劳
胃炎	洋托拉唑	洋托拉唑是一种质子泵抑制剂，能够减少胃酸的分泌，缓解胃部不适

## Mini-MYCIN解释器工作原理

### 1. 用户提问：诊断原因

- 用户输入症状（如“温度高”、“咳嗽”、“头痛”）。
- 系统得出诊断结果（如“流感”），并允许用户申请解释。

### 2. 推理过程回溯

- 根据知识①：“温度高”+“咳嗽” → 推导出“流感”，置信度为 0.8。
- 根据知识②：“温度高”+“头痛” → 推导出“流感”，置信度为 0.6。

### 3. 综合计算置信度

- 系统合并置信度： $0.8 + 0.6 \times (1 - |0.8|) = 0.92$

### 4. 输出诊断解释

- 最终置信度：0.92。
- 系统根据推理过程逐步解释原因。

## 专家系统分类

1. **解释专家系统：**通过分析输入信息，生成解释，帮助用户理解特定现象或问题的原因。
2. **预测专家系统：**基于历史和当前数据，预测未来可能发生的情况。
3. **诊断专家系统：**快速识别问题根本原因，辅助解决复杂故障或诊断问题。
4. **设计专家系统：**生成满足特定要求的设计方案，优化设计过程。
5. **规划专家系统：**制定行动计划或策略，帮助实现既定目标。
6. **控制专家系统：**实时监控和调节复杂系统运行，保障系统稳定。
7. **优化专家系统：**优化资源配置，提升效率并降低运营成本。



# 第一个成功商用的专家系统--XCON

## 1. 背景与需求

- **20世纪70-80年代初**：计算机系统配置复杂，依赖经验丰富的技术人员手动完成，错误率高且效率低。
- **DEC公司的需求**：简化配置流程，减少错误，提升生产效率。

## 2. R1系统的诞生

- **约翰·麦克德莫特**：提出利用专家系统自动化配置的想法，得到DEC支持，启动R1系统的研发。
- **初始版本**（1978年12月）：包含约200个领域规则，但处理复杂订单能力有限，错误率较高。

## 3. 改进与验证

- **第一阶段**（持续7个月）：完善规则，简化复杂订单配置，初步实现自动化。
- **第二阶段**（持续4个月）：通过专家反馈，三倍扩展领域知识，改进规则的结构与准确性。
- **正式验证**（1979年10月-11月）：R1参与50个订单配置验证，修复规则错误后所有配置均正确。

## 4. 成果与更名

验证完成后，R1系统正式更名为XCON，成为商用专家系统的标杆。

# XCON工作原理

## 1. 核心特性

- 基于规则的专家系统，采用匹配 (match) 而非生成与测试 (generate-and-test) 的方式解决问题。
- 专注于任务领域的知识，通过对订单组件的检查与补充，确保配置完整性。

## 2. 技术实现

- **语言基础**：R1 系统使用 OPS4 实现，XCON 使用 OPS5 实现，优化了易用性。
- **规则结构**：规则由条件和动作组成，存储在生产存储器与工作存储器中，系统通过识别-行动循环执行规则。
- **数据存储**：新增第三存储器（数据库），包含 VAX 系统支持的 420 个组件的详细描述及属性。

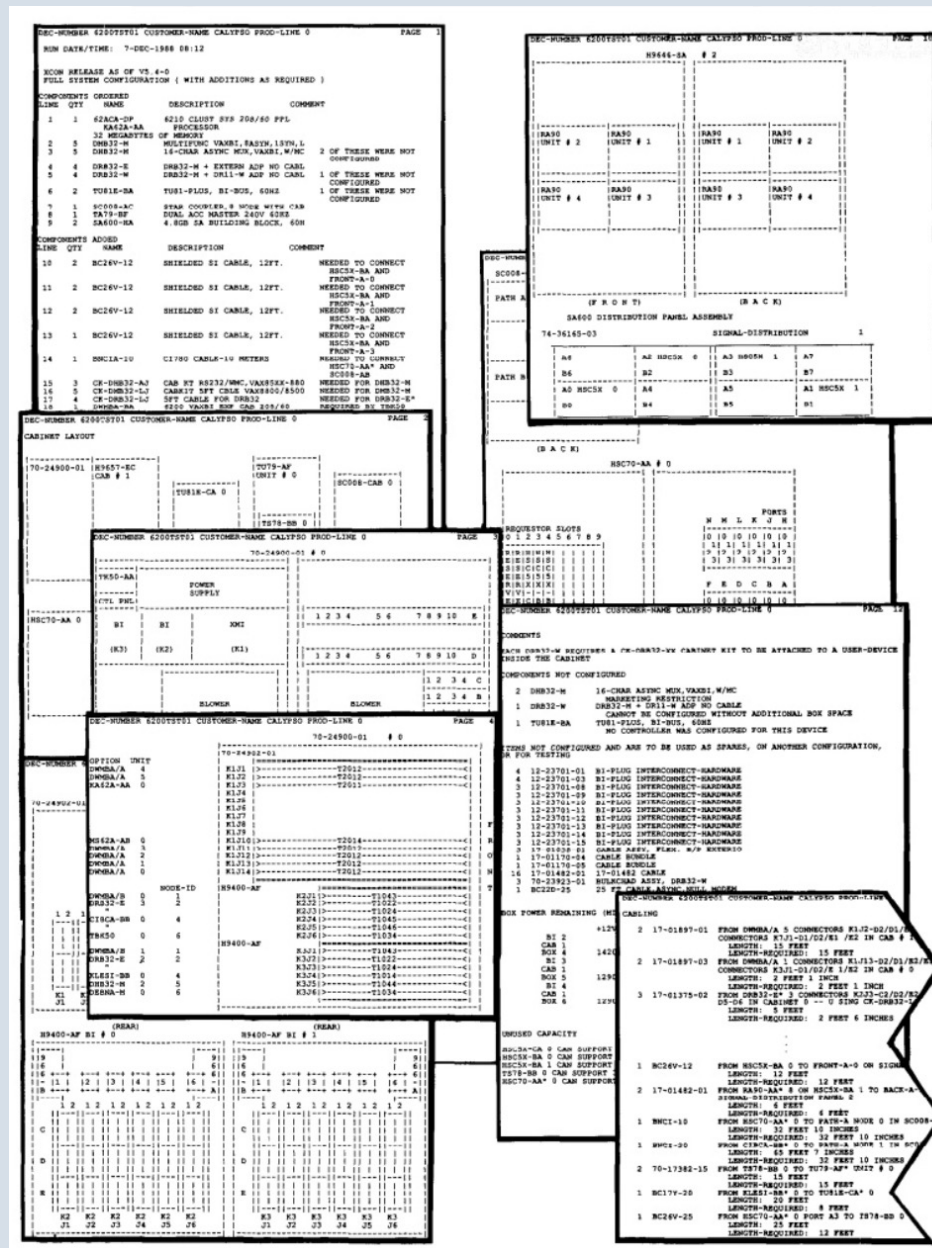
## 3. 运行流程

配置订单时，XCON 检查组件完整性，根据规则对缺失组件进行补充。

随着配置的进行，工作存储器动态增加内容，包含配置描述和任务状态。

## 4. 应用价值

提升配置效率，减少人工操作错误，广泛应用于 VAX-11/780 系统的组件配置。



## XCON规则示例

**规则名:** ASSIGN-UB-MODULES-EXCEPT-THOSE-CONNECTING-TO-PANELS-4

**IF** 当前上下文是将设备分配给UNIBUS模块  
and 存在一个未分配的双端口硬盘驱动器  
and 已知它需要的控制器类型  
and 有两个这样的控制器，它们都没有分配任何设备  
and 已知这些控制器可以支持的设备数量  
**THEN** 将硬盘驱动器分配给每个控制器  
and 注明这两个控制器已经关联  
and 每个控制器支持一个设备

**ASSIGN-UB-MODULES-EXCEPT-THOSE-CONNECTING-TO-PANELS-4**

**IF: THE CURRENT CONTEXT IS ASSIGNING DEVICES  
TO UNIBUS MODULES  
AND THERE IS AN UNASSIGNED DUAL PORT DISK DRIVE  
AND THE TYPE OF CONTROLLER IT REQUIRES IS KNOWN  
AND THERE ARE TWO SUCH CONTROLLERS NEITHER  
OF WHICH HAS ANY DEVICES ASSIGNED TO IT  
AND THE NUMBER OF DEVICES THAT THESE  
CONTROLLERS CAN SUPPORT IS KNOWN**

**THEN: ASSIGN THE DISK DRIVE TO EACH OF THE CONTROLLERS  
AND NOTE THAT THE TWO CONTROLLERS HAVE BEEN  
ASSOCIATED AND THAT EACH SUPPORTS ONE DEVICE**

## XCON规则示例

### 规则名：PUT-UB-MODULE-6

**IF** 当前上下文是将UNIBUS模块放入某个80X中的背板  
and 已确定要尝试放入背板的模块  
and 该模块是一个多路复用器终端接口  
and 它尚未与任何面板空间关联  
and 已知它所需的背板插槽类型和数量  
and 在适当类型的背板中至少有足够的插槽可用  
and 已知当前背板上的UNIBUS负载  
and 已知背板在箱子中的位置  
**THEN** 进入验证多路复用器面板空间的上下文

#### PUT-UB-MODULE-6

**IF: THE CURRENT CONTEXT IS PUTTING UNIBUS MODULES  
IN BACKPLANES IN SOME BOX  
AND IT HAS BEEN DETERMINED WHICH MODULE TO TRY  
TO PUT IN A BACKPLANE  
AND THAT MODULE IS A MULTIPLEXER TERMINAL INTERFACE  
AND IT HAS NOT BEEN ASSOCIATED WITH ANY PANEL SPACE  
AND THE TYPE AND NUMBER OF BACKPLANE SLOTS  
IT REQUIRES IS KNOWN  
AND THERE ARE AT LEAST THAT MANY SLOTS AVAILABLE  
IN A BACKPLANE OF THE APPROPRIATE TYPE  
AND THE CURRENT UNIBUS LOAD ON THAT BACKPLANE  
IS KNOWN  
AND THE POSITION OF THE BACKPLANE IN THE BOX IS KNOWN**

**THEN: ENTER THE CONTEXT OF VERIFYING PANEL SPACE  
FOR A MULTIPLEXER**



# XCON系统分析

## 1. 用户界面 (User Interface)

系统与用户之间的交互窗口，用于接收用户输入的配置需求和参数，并展示最终配置结果。

## 2. XSEL 子系统

负责选择符合需求的组件组合，利用知识库规则筛选最优配置，确保符合技术规范 and 用户要求。

## 3. XCON 子系统

系统的核心处理器，根据规则和逻辑完成组件正确组合，最终生成完整的系统配置。

## 4. 运行时库 (Run-Time Library)

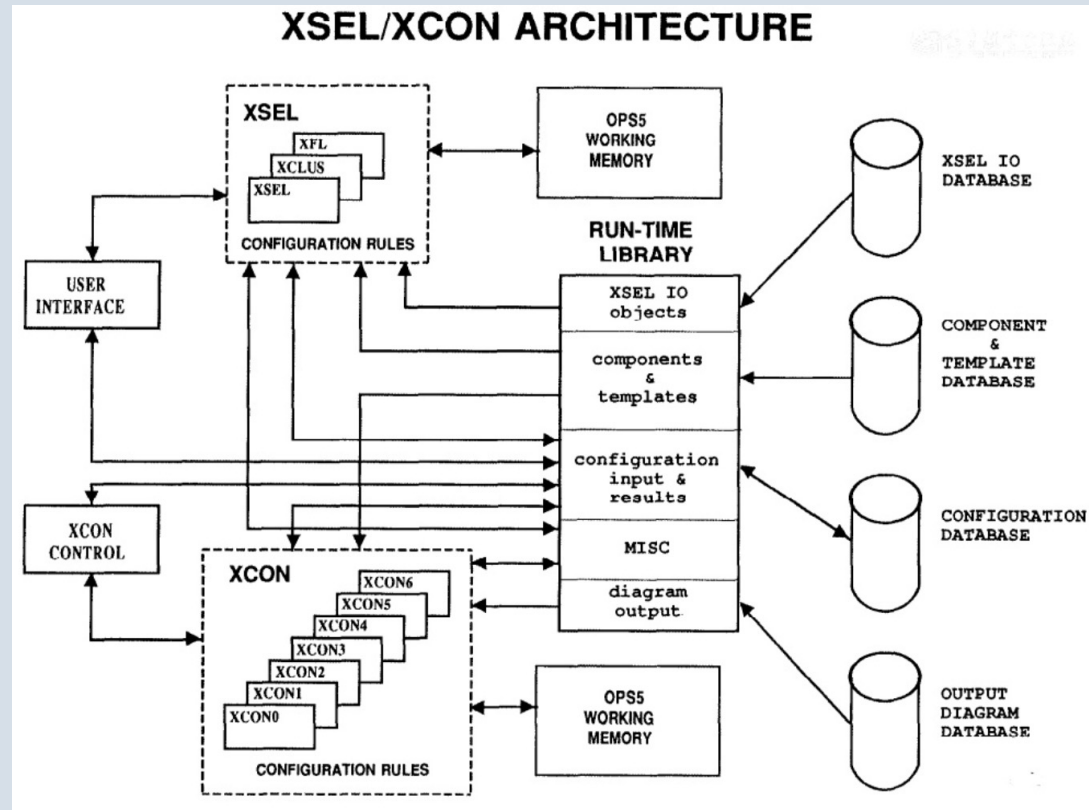
存储配置过程中所需的各种资源和信息，支持配置输入、输出及图示化展示。

## 5. OPS5 工作记忆

存储推理过程中产生的中间数据，用于支持复杂的配置决策。

## 6. 数据库组件

- **XSEL IO 数据库**：存储输入/输出数据，支持配置策略。
- **组件和模板数据库**：记录组件及模板信息，为组件选择提供支持。
- **配置数据库**：记录配置结果，便于分析和后续检查。
- **输出图示数据库**：存储配置完成后的图示结果，便于系统使用。



## 专家系统的局限性

### 1. 知识获取的瓶颈

专家系统核心是知识库，获取领域专家的隐性知识并转化为显性知识需耗费大量时间和精力。专家与系统开发人员在理解上的不对称性往往导致知识转化效率低下且不完整。

### 2. 知识库的有限性

知识库依赖于领域专家输入，范围和深度有限，无法自动更新或扩展到未知领域。知识库的更新需频繁维护，但过程耗时且无法及时反映最新研究成果。

### 3. 领域限制与通用性不足

专家系统通常针对特定领域设计，跨领域应用能力较弱，通用性和灵活性不足。在面对多样化和复杂问题时，系统显得僵化且难以应对。

## 本章小结

### 1. GOFAI 的发展历程

**定义：**GOFAI（传统人工智能）主要依赖基于规则的系统和逻辑推理，通过规则库模拟人类决策过程。

**影响：**20世纪50-70年代，推动AI领域的理论创新和实践探索，如图灵测试和约翰·麦卡锡的贡献。

### 2. 专家系统的崛起

**特点：**专家系统存储大量领域知识和推理规则提供专业咨询与解决方案。

**代表：**MYCIN 和 XCON 是具有代表性的专家系统，展示了其实际应用潜力。

### 3. 局限与启示

**局限性：**

- GOFAI 处理复杂、不确定任务能力有限，适应动态环境不足。
- 专家系统依赖知识库的维护与更新，难以扩展通用性。

**现代启示：**

- GOFAI 的理论和方法为现代AI研究奠定了基础，特别是在机器学习兴起之前。