高级人工智能——符号主义复习

——陈若愚

——2022.1.2

逻辑（Logics）：逻辑是表示信息以便得出结论的形式语言

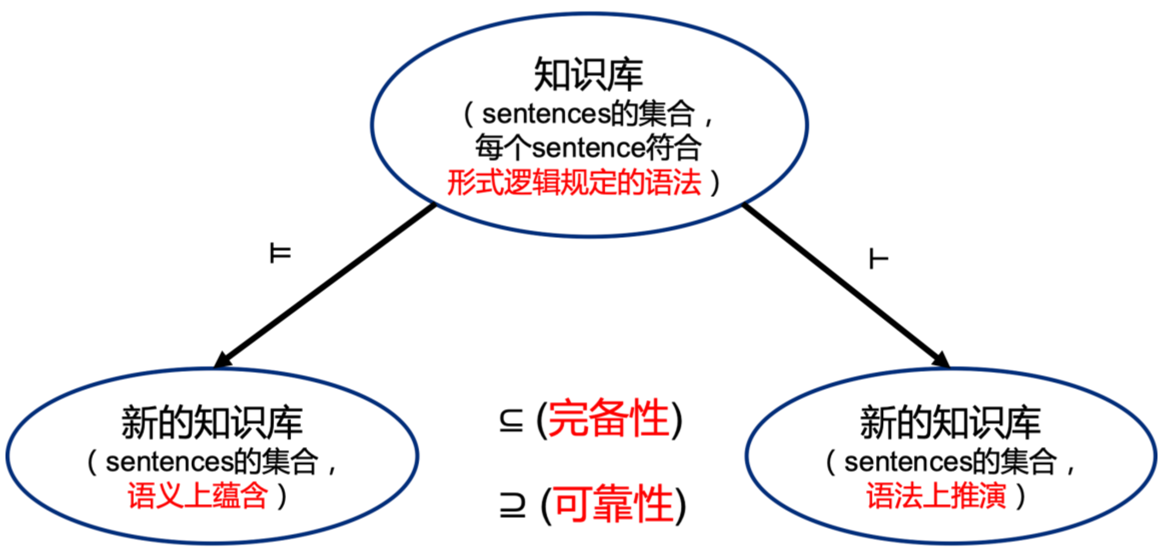
语法（Syntax）：语法定义语言中的句子

语义（Semantics）：定义句子的意思

逻辑研究的内容：研究形式化定义的 sentences 之间的关系

两个角度：

- 语义：entailment 蕴含，逻辑推导
- 语法：inference 演绎，形式推演



语义 Entailment 蕴含

蕴涵（Entailment, ⊨）意味着一件事接另一件事：

$$KB \vDash \alpha$$

知识库KB蕴涵句子α，当且仅当**α在KB为真的所有世界（代指 Model）中为真**。（这个定义要记住）

E.g., KB 包括 “the Giants won”和“the Reds won”，蕴含“Either the Giants won or the Reds won”。

E.g., $x + y = 4$蕴涵$4 = x + y$

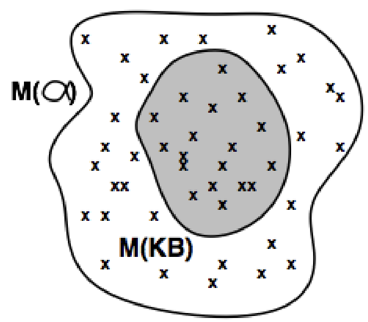
蕴涵是**基于语义**的句子(即句法)之间的一种关系

Model：如果α在模型m中为真，我们说m是句子α的模型。

E.g., X+Y=4，则 X=0，Y=4 为这个句子的 model。

$M(\alpha)$为α所有模型的集合。

重点：当且仅当$\boldsymbol{M(KB) \subseteq M(\alpha)}$时，$\boldsymbol{KB \vDash \alpha}$。



命题逻辑：语法（Syntax）与语义（Semantics）

语法（Syntax）：

命题（Proposition）：一个陈述句，要么是对的，要么是错的。

原子命题（Atomic propositions）：最小的命题。

文字（Literals）：原子命题或它们的否定。

语义（Semantics）：

每个模型指定每个命题符号的真/假

$\neg S$	is true iff	S	is false	
$S_1 \wedge S_2$	is true iff	S_1	is true and	S_2 is true
$S_1 \vee S_2$	is true iff	S_1	is true or	S_2 is true
$S_1 \Rightarrow S_2$	is true iff	S_1	is false or	S_2 is true
	i.e., is false iff	S_1	is true and	S_2 is false
$S_1 \Leftrightarrow S_2$	is true iff	$S_1 \Rightarrow S_2$	is true and	$S_2 \Rightarrow S_1$ is true

注：

$$\vDash \text{不是命题中的合法句子}$$
$$M(P \wedge Q) = M(P) \cap M(Q)$$
$$M(P \vee Q) = M(P) \cup M(Q)$$

命题逻辑中的知识库KB：

- KB：满足命题逻辑语法的 sentence 的集合；
- 假设：这组 sentence 中，一共有n个原子命题；
- 真值指派（truth assignment）：对每个原子命题赋值；
- 一共有2^n种真值指派，其中：使得**KB中的每个 sentence** 都为真的真值指派，就是**KB的 model**；
- 在此基础上，在命题逻辑中，我们可以明确的定义：

$$KB \vDash \alpha$$

Entailment ($\vDash$): 逻辑上的概念，刻画两组 sentence 之间的关系；

Implication ($\neg, \wedge, \vee, \Longrightarrow, \Longleftrightarrow$): Proposition（命题）之间的一种运算子，用真值表刻画语义。

Validity and satisfiability

- A sentence is valid if it is true in all models,
  e.g., $True,\ A \vee \neg A,\ \ A \Longrightarrow A,\ (A \wedge (A \Rightarrow B)) \Longrightarrow B$
- Validity is connected to inference via the Deduction Theorem:
  $KB \vDash \alpha$ if and only if $(KB \Longrightarrow \alpha)$ is valid
- A sentence is satisfiable if it is true in some model
  e.g., $A \vee B,\ \ C$
- A sentence is unsatisfiable if it is true in no models
  e.g., $A \wedge \neg A$
- Satisfiability is connected to inference via the following:
  $KB \vDash \alpha$ if and only if $(KB \wedge \neg \alpha)$ is unsatisfiable
- i.e., prove α by *reductio ad absurdum*

一个句子是 valid 如果他在所有模型都是 true。例如$A \vee \neg A$

证明： 当且仅在$(KB \Longrightarrow \alpha)$是 valid 情况下$KB \vDash \alpha$。

① 若$KB \vDash \alpha$，则由上文定义$M(KB) \subseteq M(\alpha)$，且$\forall m \in M(KB)$都能使$KB$与$\alpha$为真。所以此时，能保证$KB$为 True 的 model m下α为真，而不能保证KB为 True 的 model m下α不确定。即KB=True, α=True，而当KB=False 时，α为 Unknown，便是无论m为什么，$KB \Longrightarrow \alpha$，就是$(KB \Longrightarrow \alpha)$是 valid。

② 若$(KB \Longrightarrow \alpha)$是 valid，则$\forall m$使$KB$为 True 情况下，$\alpha$必然为 True。$\forall m$使$KB$为 True 的集合我们用$M(KB)$表示，$\forall m \in M(KB)$都能使$KB$与$\alpha$为真，故$M(KB) \subseteq M(\alpha)$，由于这是$KB \vDash \alpha$的充分必要条件，所以$KB \vDash \alpha$。

故$(KB \Longrightarrow \alpha)$是 valid 与$KB \vDash \alpha$互为充分必要条件。

证毕。

证明： 只有在$KB \wedge \neg\alpha$为 unsatisfiable 时候，$KB \vDash \alpha$。（就是任何m都不能满足$KB \wedge \neg\alpha$为 True）

① 假设存在m可以使$KB \wedge \neg\alpha$为 True 且$KB \vDash \alpha$，则必然使KB为 True，而α为 False。若$KB \vDash \alpha$，则$\forall m \in M(KB)$，使KB为真，此条件下必然使α也为真，而与上一步中KB为 True 但α为 False 的结论冲突，故假设不成立。不存在任何m可以使$KB \wedge \neg\alpha$和$KB \vDash \alpha$同时成立。

② 当$KB \wedge \neg\alpha$为 unsatisfiable 时候，$\forall m$使KB成立，必然使α成立，而这满足仅当**$\boldsymbol{\alpha}$在$\boldsymbol{KB}$为真的所有 model 中为真**时，$KB \vDash \alpha$成立。

故只有在$KB \wedge \neg\alpha$为 unsatisfiable 时候，$KB \vDash \alpha$。

证毕。

总结：蕴含的三个等价条件：

$$KB \vDash \alpha$$

$$M(KB) \subseteq M(\alpha)$$

$$KB \Longrightarrow \alpha \text{是 valid（即永真）}$$

$$KB \wedge \neg\alpha \text{为 unsatisfiable（即永假）}$$

形式推演（Deduction, $\vdash$）：

符号$\vdash$表示可推演，设$\Sigma = \{A_1, A_2, \dots, A_n\}$，若$A$由形式$\Sigma$可推演，则可以表示为：

$$\Sigma \vdash A$$

Soundness（可靠性）: if $KB \vdash_i \alpha$ then $KB \vDash \alpha$

Completeness（完备性）: if $KB \vDash \alpha$ then $KB \vdash_i \alpha$

归结原理（Resolution）

合取范式：Conjunctive Normal Form (CNF—universal)

目的是将一些列命题用∧和∨连起来，具体步骤如下：

- $B_{1,1} \Leftrightarrow (P_{1,2} \lor P_{2,1})$
- 1. Elimate⇔,replacing $\alpha \Leftrightarrow \beta$ with $(\alpha \Rightarrow \beta) \land (\beta \Rightarrow \alpha)$

  $(B_{1,1} \Rightarrow (P_{1,2} \lor P_{2,1})) \land ((P_{1,2} \lor P_{2,1}) \Rightarrow B_{1,1})$
- 2.Elimate ⇒ ,replacing $\alpha \Rightarrow \beta$ with $\neg\alpha \lor \beta$.

  $(\neg B_{1,1} \lor P_{1,2} \lor P_{2,1}) \land (\neg(P_{1,2} \lor P_{2,1}) \lor B_{1,1})$
- 3.Move ¬ inwards using de Morgan's rules and double-negation

  $(\neg B_{1,1} \lor P_{1,2} \lor P_{2,1}) \land ((\neg P_{1,2} \land \neg P_{2,1}) \lor B_{1,1})$
- 4.Apply distributivity law (∨ over ∧) and flatten:

  $(\neg B_{1,1} \lor P_{1,2} \lor P_{2,1}) \land (\neg P_{1,2} \lor B_{1,1}) \land (\neg P_{2,1} \lor B_{1,1})$

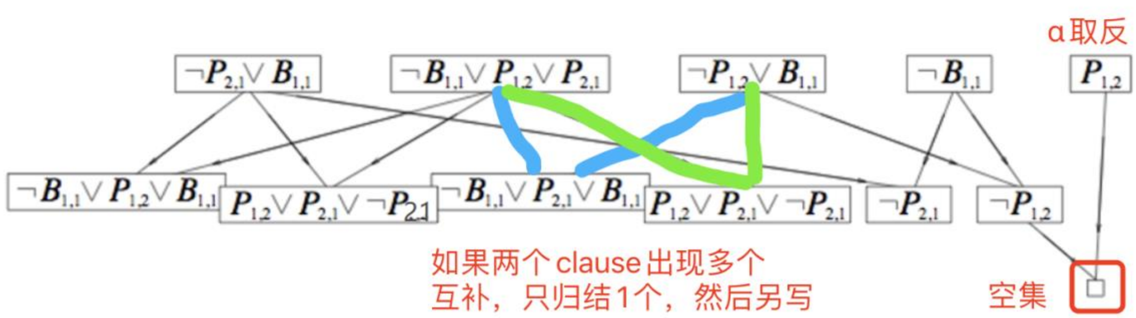
注意：$A \Rightarrow B$等价于$\neg A \lor B$

归结：

$$\frac{P_{1,3} \lor P_{2,2}, \qquad \neg P_{2,2}}{P_{1,3}}$$

一般要证明KB ⊢ α，通常需要使$\{KB, \neg\alpha\} \vdash \emptyset$，对$\alpha$要取反，然后归结如果找到空集结论成立，如果找不到空集，则结论不成立，例子：

$$KB = (B_{1,1} \Leftrightarrow (P_{1,2} \lor P_{2,1})) \land \neg B_{1,1} \qquad \alpha = \neg P_{1,2}$$



$$KB = (B_{1,1} \Leftrightarrow (P_{1,2} \lor P_{2,1})) \land \neg B_{1,1}$$

$$[B_{1,1} \Rightarrow (P_{1,2} \lor P_{2,1})] \land [(P_{1,2} \lor P_{2,1}) \Rightarrow B_{1,1}] \land \neg B_{1,1}$$

$$[\neg B_{1,1} \lor P_{1,2} \lor P_{2,1}] \land [(\neg P_{1,2} \land \neg P_{2,1}) \lor B_{1,1}] \land \neg B_{1,1}$$

$$= [\neg B_{1,1} \lor P_{1,2} \lor P_{2,1}] \land [\neg P_{1,2} \lor B_{1,1}] \land [\neg P_{2,1} \lor B_{1,1}] \land \neg B_{1,1}$$

通过归结原理证明可靠性与完备性：

可靠性（Sound）： if $KB \vdash_i \alpha$ then $KB \vDash \alpha$

就是证明这个：

$$(l_1 \lor \cdots \lor l_k) \land (m_1 \lor \cdots \lor m_n)$$
$$\vDash (l_1 \lor \cdots \lor l_{i-1} \lor l_{i+1} \lor \cdots \lor l_k \lor m_1 \lor \cdots \lor m_{j-1} \lor m_{j+1} \lor \cdots \lor m_n)$$

很简单，检查真值表即可。

Resolution规则： 分子都为真时，分母要为真就 ok 了

$$\frac{\ell_1 \lor \cdots \lor \ell_k, \qquad m_1 \lor \cdots \lor m_n}{\ell_1 \lor \cdots \lor \ell_{i-1} \lor \ell_{i+1} \lor \cdots \lor \ell_k \lor m_1 \lor \cdots \lor m_{j-1} \lor m_{j+1} \lor \cdots \lor m_n}$$

完备性（Completeness）： if $KB \vDash \alpha$ then $KB \vdash_i \alpha$

证明也非常简单，归结原理最终要求 clause 与$\neg\alpha$归结为空集，即$\{KB, \neg\alpha\} \vdash \emptyset$。

证明若$KB \land \neg\alpha$是 unsatisfiable，则$KB \vdash_i \alpha$。

令$S = \{KB, \neg\alpha\}$

因为S是 unsatisfiable，故 RC(S)包含空集

我们要证明逆否命题，**$RC(S)$不包含空集的话，则S是 satisfiable 的**。

证：由解析推导出的所有子句和 S 中的所有句子的集合的 clause 我们用$RC(S)$代替。假设 S 中的一些原子命题$R_1, R_2, \ldots, R_l$，构造如下的 model：

For i=1 to l:对$R_1, R_2, \ldots, R_l$指派真值。

若$RC(S)$中包含一个子句，此子句包含$\neg R_i$，且其他子句都被指派为 False 了（**如果其他子句不一定可以强制为 False，则可以保证子**

句为真，因为子句都是∨连接），则应该把R_i指派 False，否则指派 True。**该 Model 将会把$\boldsymbol{RC(S)}$中的子句都为真，这样是 satisfiable 的**。

这个真值指派使$RC(S)$中的子句都为真。

用反证法证明：假设在第 i 步，指派R_i使某个子句 C 为 False，且假设是第一次出现 False 子句，此时子句 C 只能是以下的两种形势：

$$\boldsymbol{False \lor False \cdots \lor False \lor R_i} \quad \textbf{或者} \quad \boldsymbol{False \lor False \cdots \lor False \lor \neg R_i}$$

但是如果$RC(S)$中包括两个子句之一，子句C是不会在此真值表中出现 False，**这是因为归结选择的问题（只能无论选择什么都变为 False，如果能是调节为 True 还会是 True**）。因此只有可能是两种情况都存在才能出现某一个句子是 False 的情况。但是两种情况都存在的话是满足归结条件的(**但之前说了$\boldsymbol{RC(S)}$已经是最终归结**)，也就是说他们归结后的子句会出现在$RC(S)$中，这是一个永假的子句，与之前的假设矛盾。故如果**$\boldsymbol{RC(S)}$不包含空集的话，则不可能把一个子句弄为永假的，所以$\boldsymbol{S}$是 Satisfiable**。

如果文字包含否定符号(¬),将其称为“负文字”(negative literal),否则称为“正文字”(positive literal)。

Definite clause：有且只有一个正文字

Horn clause：最多只有一个正文字

Modus Ponens：

$$\frac{\alpha_1,\ldots,\alpha_n, \qquad \alpha_1 \land \cdots \land \alpha_n \Rightarrow \beta}{\beta}$$

Modus Ponens 的可靠性（soundness）证明：与归结原理的可靠性证明一样，用真值表就行了。

即证明：

$$\alpha_1 \land \cdots \land \alpha_n \land (\alpha_1 \land \cdots \land \alpha_n \Rightarrow \beta) \vDash \beta$$

Modus Ponens 的完备性（completeness）证明：

即证明 if $KB \vDash \alpha$ then $KB \vdash_i \alpha$

因为 Modus Ponens 推理规则为：

$$\frac{\alpha_1,\ldots,\alpha_n, \qquad \alpha_1\wedge\cdots\wedge\alpha_n\Rightarrow\beta}{\beta}$$

首先，KB只会包含 Definite clause：有且只有一个正文字，就是上图的分子最右侧。我们假设推理的子句集合为$RC(KB)$，构建一个 model，则在该 model 下针对任意的$\alpha \in RC(KB)$都需要保证是 True。

如果假设存在 model 使KB为 False，则必须使其中一个子句为假。若子句为$\alpha_1\wedge\cdots\wedge\alpha_n \Rightarrow \beta$，其中$\alpha_i$都为真，必须使$\beta$为假，但此时因为$\beta \in RC(KB)$，$\beta$应当为真，存在矛盾，假设不成立。若子句只为$\beta$，如果其为假，但该子句$\beta \in RC(KB)$，还是存在矛盾。

若$KB \vDash \alpha$，在使KB为真的 model 下α也为真，而$\alpha \in RC(KB)$，故$KB \vdash_i \alpha$。

一阶谓词逻辑

Objects: people, house, numbers, theories, Ronald McDonald, colors, baseball games, wars, centuries…..

Relations: red, round, bogus, prime, multistoried…,
brother of, bigger than, inside, part of, has color, occurred after, owns, comes between,…

Functions: father of, best friend, third inning of, one more than, end of…

Constants *KingJohn, 2, UCB,…*
Predicates Brother, >,…
Functions *Sqrt, LeftLegOf,…*
Variables x, y, a, b,…
Connectives $\wedge$ $\vee$ $\neg$ $\Rightarrow$ $\Leftrightarrow$
Equality =
Quantifiers $\forall$ $\exists$

Universal quantification:

$\forall < variables >< sentence >$

Everyone at Berkeley is smart:

$\forall x\ At(x, Berkeley) \Rightarrow Smart(x)$

Existential quantification:

$\exists < variables >< sentence >$

Someone at Stanford is smart:

$\exists x\ At(x, Stanford) \wedge Smart(x)$

$\exists x\ \forall y\ Loves(x, y)$

"There is a person who loves everyone in the world"

$\forall y\ \exists x\ Loves(x, y)$

"Everyone in the world is loved by at least one person"

$\forall x\ Likes(x, IceCream)$ $\neg\exists x\ \neg Likes(x, IceCream)$
$\exists x\ Likes(x, Broccoli)$ $\neg\forall x\ \neg Likes(x, Broccoli)$

全称实例化 **Universal Instantiation (UI)**

$$\frac{\forall v\ \alpha}{SUBST(\{^{v}/_{g}\}, \alpha)}$$

对于任何变量v和基项g:

E.g., $\forall x\ King(x) \wedge Greedy(x) \Rightarrow Evil(x)$ yields
$King(John) \wedge Greedy(John) \Rightarrow Evil(John)$
$King(Richard) \wedge Greedy(Richard) \Rightarrow Evil(Richard)$
$King(Father(John)) \wedge Greedy(Father(John)) \Rightarrow Evil(Father(John))$

存在实例化 **Existential Instantiation (EI)**

$$\frac{\exists v\ \alpha}{SUBST(\{v/k\}, \alpha)}$$

E. g. , $\exists x\ Crown(x) \wedge OnHead(x, John)$ yields
$Crown(C_1) \wedge OnHead(C_1, John)$
Provided C_1 is a new constant symbol, called a **Skolem constant**

C1 是一个不存在 KB 中的词，skolem 是指使用过了，下次替换不可以用 C1。

Unification（合一化）

- We can get the inference immediately if we can find a substitution θ such that $King(x)$ and $Greedy(x)$ match $King(John)$ and $Greedy(y)$
- $\theta = \{x/John, y/John\}$ works
- $UNIFY(\alpha, \beta) = \theta$ if $\alpha\theta\ equals\ to\ \beta\theta$

p	q	θ
$Knows(John, x)$	$Knows(John, Jane)$	$\{x/Jane\}$
$Knows(John, x)$	$Knows(y, OJ)$	$\{x/OJ, y/John\}$
$Knows(John, x)$	$Knows(y, Mother(y))$	$\{y/John, x/Mother(John)\}$
$Knows(John, x)$	$Knows(x, OJ)$	$fail$

Standardizing apart eliminates overlap of variables, e.g., $Knows(z_{17}, OJ)$

一阶谓词逻辑的归结原理：

Full first-order version

$$\frac{l_1 \vee \cdots \vee l_k, \qquad m_1 \vee \cdots \vee m_n}{(l_1 \vee \cdots \vee l_{i-1} \vee l_{i+1} \vee \cdots \vee l_k \vee m_1 \vee \cdots \vee m_{j-1} \vee m_{j+1} \vee \cdots \vee m_n)\theta}$$

Where $UNIFY(l_i, \neg m_j) = \theta$

For example,

$$\frac{\neg Rich(x) \vee Unhappy(x)\,, \qquad Rich(Ken)}{Unhappy(Ken)}$$

With $\theta = \{x/Ken\}$

Apply resolution steps to CNF(KB$\wedge \neg\alpha$); complete for FOL

举例：

Everyone who loves animals is loved by someone:
$\forall x\ [\forall y\ Animal(y) \Rightarrow Loves(x,y)] \Rightarrow [\exists y\ \ Loves(y,x)]$

Eliminate biconditionals and implications $\forall x \neg [\forall y\ \neg Animal(y) \vee Loves(x,y)] \vee [\exists y\ \ Loves(y,x)]$

Move $\neg$ inwards: $\neg \forall\ x,p\ \ \equiv \exists x\ \neg p,\ \ \neg \exists x,p\ \ \equiv \forall x\ \neg p$:

$$\forall x\ \ [\exists y\ \neg(\neg Animal(y) \vee Loves(x,y))] \vee [\exists y\ \ Loves(y,x)]$$
$$\forall x\ \ [\exists y\ \neg\neg Animal(y) \wedge \neg Loves(x,y))] \vee [\exists y\ \ Loves(y,x)]$$
$$\forall x\ \ [\exists y\ Animal(y) \wedge \neg Loves(x,y))] \vee [\exists y\ \ Loves(y,x)]$$

Standardize variables: each quantifier should use a different one

$$\forall x\ \ [\exists y\ Animal(y) \wedge \neg Loves(x,y))] \vee [\exists z\ \ Loves(z,x)]$$

Skolemize: a more general form of existential instantiation.
Each existential variable is replaced by a Skolem function of the enclosing universally quantified variables:

$$\forall x\ \ [\ Animal(F(x)) \wedge \neg Loves(x,F(x)))] \vee [\ Loves(G(x),x)]$$

Drop universal quantifiers:

$$[\ Animal(F(x)) \wedge \neg Loves(x,F(x))] \vee [\ Loves(G(x),x)]$$

Distribute $\wedge$ over $\vee$:

$$[\ Animal(F(x)) \vee Loves(G(x),x))] \wedge [\neg Loves(x,F(x)) \vee Loves(G(x),x)]$$

归结策略：广度优先

- **广度优先策略的优点：**
  - 当问题有解时保证能找到最短归结路径。
  - 是一种完备的归结策略。
- **广度优先策略的缺点：**
  - 归结出了许多无用的子句
  - 既浪费时间，又浪费空间

- **广度优先对大问题的归结容易产生组合爆炸，但对小问题却仍是一种比较好的归结策略。**

模糊逻辑表达

模糊性：事件发生的程度，而不是是否发生
随机性：事情发生的不确定性

- 定义 设**U**是给定论域，μ_F是把任意**u**∈**U**映射为**[0，1]**上某个实值的函数，即

$$\mu_F \ : \ \mathbf{U} \to [\mathbf{0}, \mathbf{1}]$$

- 则称 μ_F为定义在**U**上的一个隶属函数，由 $\mu_F(u)$（对所有$u \in U$）所构成的集合**F**称为**U**上的一个模糊集，$\mu_F(u)$称为**u**对**F**的隶属度。

- 模糊集F完全是由隶属函数μ_F来刻画的，μ_F把U中的每一个元素u都映射为[0, 1]上的一个值$\mu_F(u)$ 。
- $\mu_F(u)$ 的值表示u隶属于F的程度，其值越大，表示u隶属于F的程度越高。当 $\mu_F(u)$仅取0和1时，模糊集F便退化为一个普通集合。

模糊集的表示

为了能够表示出论域中的元素与其隶属度之间的对应关系，扎德引入了一种模糊集的表示方式：先为论域中的每个元素都标上其隶属度，然后再用+号把它们连接起来：

$$F = \mu_F(u_1)/u_1 + \mu_F(u_2)/u_2 + \cdots + \mu_F(u_n)/u_n$$

连续论域的表示方法

- 如果论域是连续的，则其模糊集可用一个实函数来表示。
  - 例如，扎德以年龄为论域，取U=[0, 100]，给出了“年轻”与“年老”这两的模糊概念的隶属函数

$$\mu_{年老}(u) = \begin{cases} 0 & 当 0 \le u \le 50 \\ [1 + (\frac{5}{u-50})^2]^{-1} & 当 50 < u \le 100 \end{cases} \qquad \mu_{年轻}(u) = \begin{cases} 1 & 当 0 \le u \le 25 \\ [1 + (\frac{u-25}{5})^2]^{-1} & 当 25 < u \le 100 \end{cases}$$

一般表示方法

- 不管论域U是有限的还是无限的，是连续的还是离散的，扎德又给出了一种类似于积分的一般表示形式：

$$F = \int_{u \in U} \mu_F(u)/u$$

- 这里的记号不是数学中的积分符号，也不是求和，只是表示论域中各元素与其隶属度对应关系的总括。

模糊集运算：

定义　设F、G分别是U 上的两个模糊集，对任意u∈U，都有$\mu_F(u) = \mu_G(u)$成立，则称F等于G，记为F=G。

设F、G分别是U上的两个模糊集，对任意u∈U，都有$\mu_F(u) \le \mu_G(u)$成立，则称F包含G，记为F ⊆ G。

定义　设F、G分别是U上的两个模糊集，则F⋃G、F⋂G分别称为F与G的并集、交集，它们的隶属函数分别为：

$$F \cup G: \mu_{F \cup G}(u) = \max_{u \in U}\{\mu_F(u), \mu_G(u)\}$$
$$F \cap G: \mu_{F \cap G}(u) = \min_{u \in U}\{\mu_F(u), \mu_G(u)\}$$

定义　设F为U上的模糊集，称﹁F为F的补集，其隶属函数为：

$$\neg F: \mu_{\neg F}(u) = 1 - \mu_F(u)$$

两个模糊集之间的运算实际上就是逐点对隶属函数作相应的运算 。

“又矮又瘦”

- U = {甲, 乙, 丙, 丁}

  A = “矮子”　隶属函数 (0.9, 1, 0.6, 0)

  B = “瘦子”　隶属函数 (0.8, 0.2, 0.9, 1)
- 找出 C = “又矮又瘦”
- C = A∩B　= (0.9∧0.8 , 1∧0.2 , 0.6∧0.9 , 0∧1)

  = (0.8, 0.2, 0.6, 0)
- 因此， 甲和丙比较符合条件

模糊关系定义：

- 定义 设F_i是U_i(i=1, 2, …, n)上的模糊集，则称

$$F_1 \times F_2 \times \cdots \times F_n = \int_{u_1 \times u_2 \times \cdots \times u_n} (\mu_{F_1}(u_1) \wedge \mu_{F_2}(u_2) \wedge \cdots \wedge \mu_{F_n}(u_n))/(u_1, u_2, \cdots, u_n)$$

为$F_1, F_2, \ldots, F_n$的笛卡尔乘积，它是$U_1 \times U_2 \times \ldots \times U_n$ 上的一个模糊集。

- 定义 在$U_1 \times U_2 \times \cdots \times U_n$上的一个n元模糊关系R是指以$U_1 \times U_2 \times \cdots \times U_n$为论域的一个模糊集，记为

$$R = \int_{U_1 \times U_2 \times \cdots \times U_n} \mu_R(u_1, u_2, \cdots, u_n) \,/(u_1, u_2, \cdots, u_n)$$

例 设有一组学生 U={u_1,u_2}={秦学，郝玩}，一些在计算机上的活动V={v_1,v_2,v_3}={编程，上网，玩游戏}

并设每个学生对各种活动的爱好程度分别为 $\mu_F(u_i, v_j)$ **i=1,2；j=1,2,3，即**

μ_R(秦学, 编程) = 0.9, μ_R(秦学, 上网) = 0.6，μ_R(秦学, 玩游戏) = 0,
μ_R(郝玩, 编程) = 0.2, μ_R(郝玩, 上网) = 0.3，μ_R(郝玩, 玩游戏) = 0.8

则U×V上的模糊关系R为

$$R = \begin{bmatrix} 0.9 & 0.6 & 0 \\ 0.2 & 0.3 & 0.8 \end{bmatrix}$$

模糊关系的合成（这个重点关注下）：

定义 设$\mathbf{R}_1$与$\mathbf{R}_2$分别是$\mathbf{U}\times\mathbf{V}$与$\mathbf{V}\times\mathbf{W}$上的两个模糊关系，则$\mathbf{R}_1$与$\mathbf{R}_2$的合成是从$\mathbf{U}$到$\mathbf{W}$的一个模糊关系，记为 $\mathbf{R}_1\mathbf{o}\mathbf{R}_2$。其隶属函数为

$$\mu_{R_1\circ R_2}(u,w)=\vee\{\mu_{R_1}(u,v)\wedge\mu_{R_2}(v,w)\}$$

其中，∧和∨分别表示取最小和取最大。

模糊关系合成举例

- 例 设有：
- 一组学生 $U=\{u_1,u_2\}$={秦学，郝玩}，
- 一些在计算机上的活动$V=\{v_1,v_2,v_3\}$={编程，上网，玩游戏}
- 一些对学生的评价$G=\{g_1,g_2\}$={好，差}
- 若已知U和V的模糊关系，V和G的模糊关系，
- 那么，我们就可以合成出U和G的模糊关系

模糊关系的合成

- 例 设有以下两个模糊关系

$$R_1=\begin{bmatrix}0.4 & 0.5 & 0.6\\ 0.8 & 0.3 & 0.7\end{bmatrix}\qquad R_2=\begin{bmatrix}0.7 & 0.9\\ 0.2 & 0.8\\ 0.5 & 0.3\end{bmatrix}$$

- 则R_1与R_2的合成是

$$R=R_1\circ R_2=\begin{bmatrix}0.5 & 0.5\\ 0.7 & 0.8\end{bmatrix}$$

类比矩阵乘法

- 把R_1的第i行元素分别与R_2的第j列的对应元素相比较，两个数中取最小者，然后再在所得的一组最小数中取最大的一个，并以此数作为R_1oR_2的元素R(i,j)。