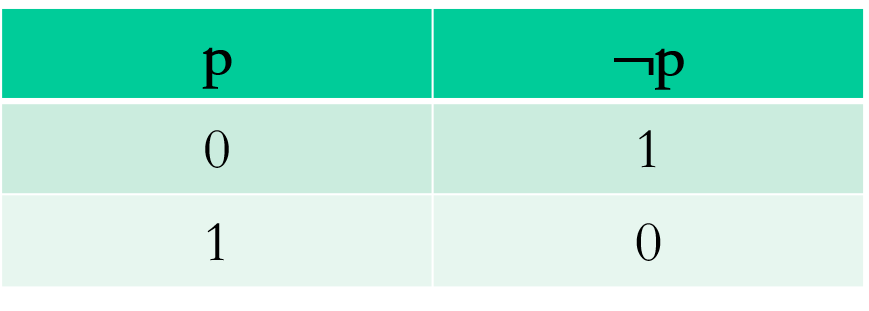
**逻辑联结词—非(¬)**

* **命题p的否定记为**¬**p,读作非p**

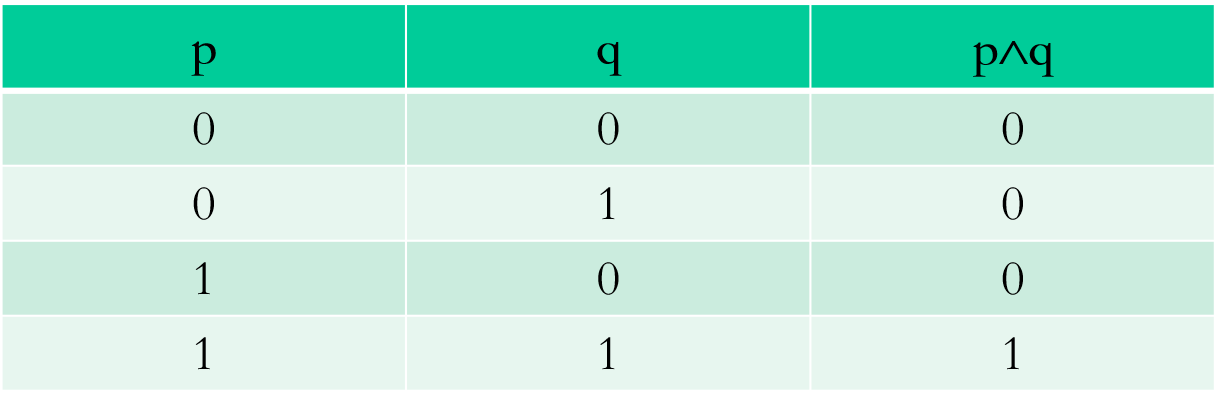
**设p为一命题，p的否定为一个新的命题**



**逻辑联结词— 合取(∧)**

* **p**∧**q称为p和q的合取。相当于“并且” 。**

**设p, q为命题，则p且q，或 p合取q 为真当且仅当p和q同时为真，否则为假。**



Ex:

* **含义与自然语言不同**

**p：今天下雨了**

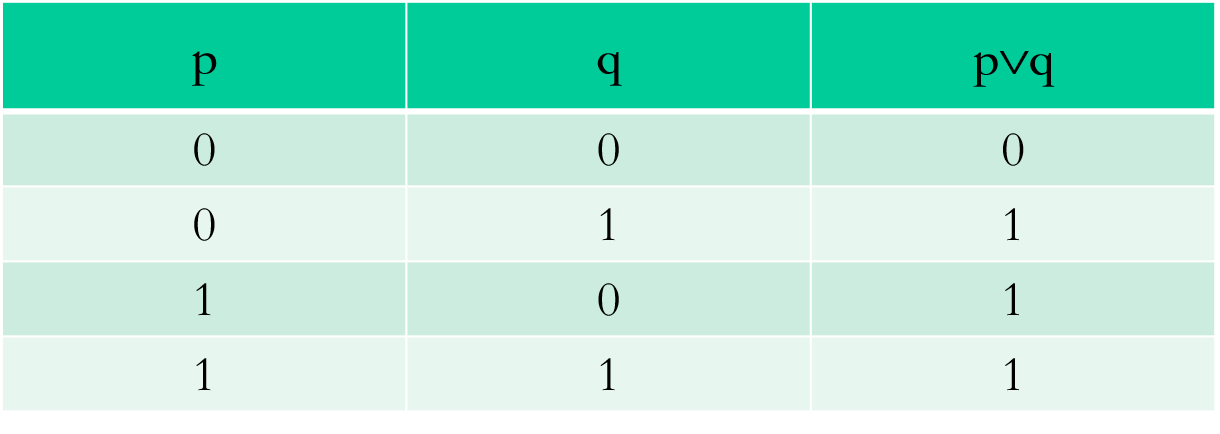
**q：会议室有人在开会**

* **p**∧**q 为数理逻辑中的一新命题，具有真假值**
* **在自然语言中，“今天下雨了且会议室有人在开会”没有意义。**

**逻辑联结词— 析取(∨)**

* **p**∨**q称为p和q的析取。相当于“或” 。**

**设p, q为命题，则p或q，或 p析取q 为假当且仅当p和q同时为假，否则为真。**



* **含义不同于自然语言**

**r：今天晚上我在家看电视或去剧场看戏**

**s：我若今天回来得早或不累就去找你**

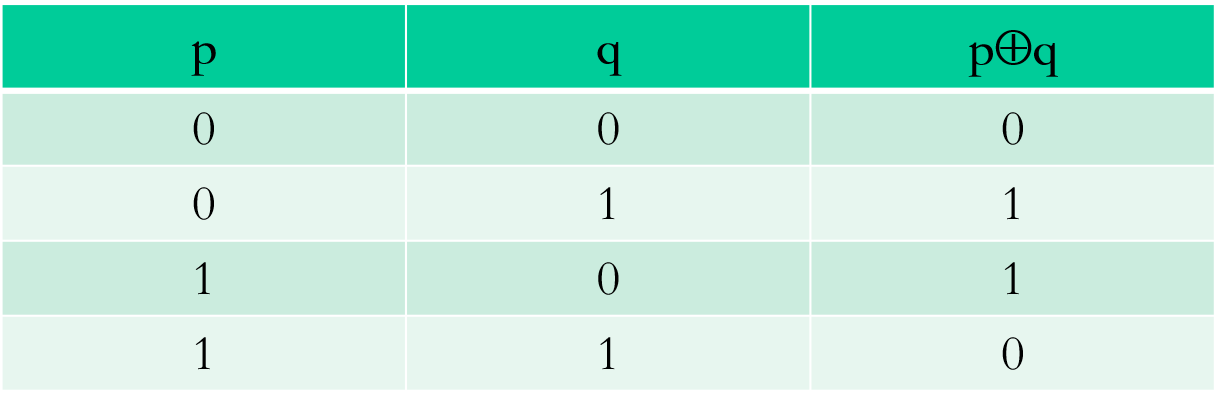
* **自然语言中，r为排斥或，s为可兼或。**
* **在数理逻辑中，析取指的是s，可兼或。**

**逻辑联结词— 异或(⊕)**

* **p**⊕**q称为p和q的异或。相当于“或” 。**

**设p, q为命题，则p异或q为真当且仅当p和q不**

**同，否则为假。**



* **含义不同于自然语言**

**r：今天晚上我在家看电视或去剧场看戏**

**s：我若今天回来得早或不累就去找你**

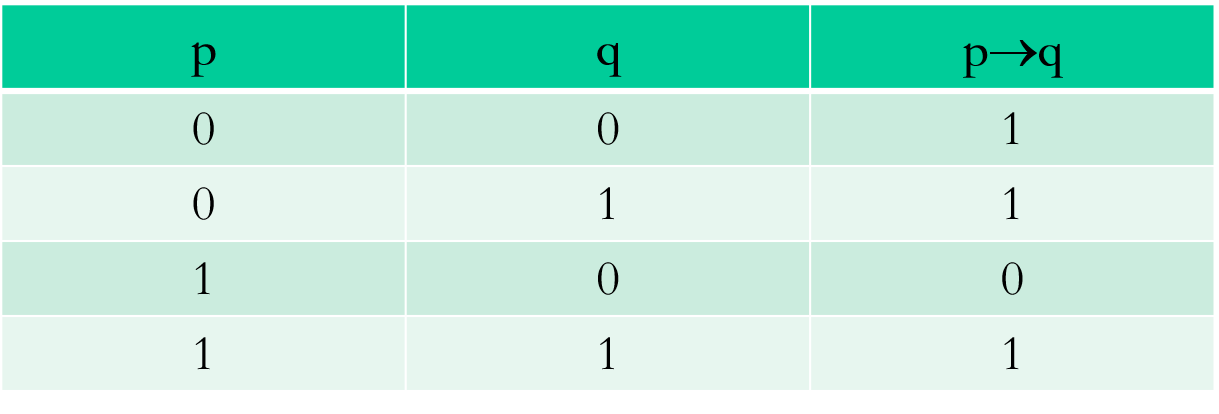
* **自然语言中，r为排斥或，s为可兼或。**
* **在数理逻辑中，异或指的是r，排斥或。**

**逻辑联结词— 蕴含(→)**

* **p**→**q称为p蕴含q。相当于“如果…则…” 。**

**设p, q为命题，则p蕴含q为假当且仅当p为真且q为假**

**否则为真。P称为前件(条件)，q为后件(结论)。**



* **含义不同于自然语言**

**r：如果太阳从西边出来，则雪是黑的。**

**s：如果明天下雨，则不开运动会。**

**t：如果我今天死，则我长生不老。**

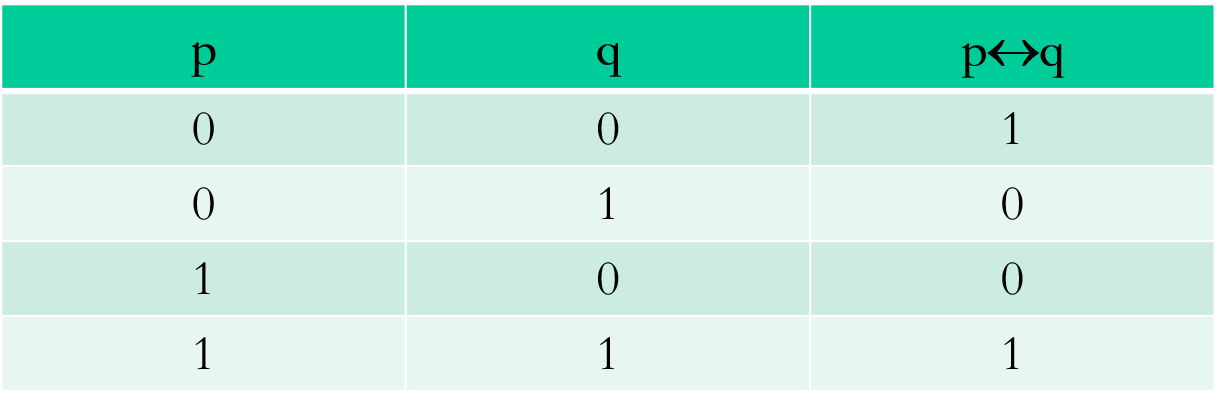
* **自然语言中，r，t是无意义的。**
* **在数理逻辑中，r，s，t都可表示，且r,t是真命题. 蕴含称为实质蕴含。**

**逻辑联结词— 等价(↔)**

* **p**↔**q称为p等价于q。相当 “当且仅当iff” 。**

**设p, q为命题，则 p等价于q 为真当且仅当p 与q真值**

**相等，否则为假。**



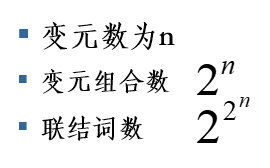
* **含义不同于自然语言**

**r：2+2=4当且仅当雪是白的。**

**s：两个三角形全等，当且仅当它们的三组对**

**应边相等。**

* **自然语言中，r是无意义的。**
* **在数理逻辑中，r，s都可表示。**



Example:

* **李明是计算机系的学生，他住在312室或313室。**

**p：李明是计算机系的学生**

**q：他住在312室**

**r：他住在313室**

**p ∧ (q ⊕ r)**

* **燕子飞回来是春天来了的必要条件。**

**p：燕子飞回来**

**q：春天来了**

**q →p**

* **如果我下班早且不累，就去商店看看。**

**p：我下班早**

**q：我累了**

**r：我去商店看看**

**(p ∧ (¬ q ))→r**

* **如果明天下雨，就不开运动会而照常上课。**

**p：明天下雨**

**q：明天开运动会**

**r：明天照常上课**

**p →((¬ q）∧ r )**

* **二元联结词** ∧ **（合取），** ∨ **（析取），** ⊕ **（异或），** → **（蕴涵），** ↔ **（等价）的真值表如下。**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| *p* | *q* | *p* ∧ *q* | *p* ∨ *q* | *p* ⊕ *q* | *p* → *q* | *p* ↔ *q* |
| 0 | 0 | 0 | 0 | 0 | 1 | 1 |
| 0 | 1 | 0 | 1 | 1 | 1 | 0 |
| 1 | 0 | 0 | 1 | 1 | 0 | 0 |
| 1 | 1 | 1 | 1 | 0 | 1 | 1 |

**(命题逻辑)合式公式定义**

* **(1) 常元0和1是合式公式；**
* **(2) 命题变元是合式公式；**
* **(3) 若Q, R是合式公式，则：**
  1. **(**¬**Q)、(Q** ∧ **R) 、(Q** ∨ **R) 、**
  2. **(Q**→**R) 、(Q** ↔ **R) 、(Q** ⊕ **R)是合式公式；**
* **(4) 只有有限次应用规则(1)—(3)构成的公式是合式公式。**

**合式公式**

* **设S是联结词的集合。由S生成的公式定义如下：**
  1. **1）若c是S中的0元联结词，则c是由S生成的公式。**
  2. **2）原子公式是由S生成的公式。**
  3. **3）若n≥1，F是S中的n元联结词，A1,…,An是由S生成的公式，则FA1…An是由S生成的公式。**
* **卢卡西维茨运算符表示法**
  1. **前缀表示，FA1…,An;**
  2. **中缀表示，A1FA2;**
  3. **后缀表示，A1…,AnF**

**真值赋值**

* **由S生成的公式Q在真值赋值v下的真值v(Q)定义如下：**
* **⑴若Q是S中的0元联结词c，则v(Q)=c。**
* **⑵若Q是命题变元p，则v(Q)= pv。**
* **⑶若Q是FQ1…,Qn，其中n≥1， F是S中的n元联结词， Qi是公式，则v(Q)=v(FQ1…Qn)=Fv(Q1)…v(Qn)。**

**可满足性和有效性**

* **设Q是公式。**
* **⑴如果真值赋值v使得v(Q)=1，则称v满足Q。**
* **⑵如果每个真值赋值都满足Q，则称Q为有效式，或称为永真式，也称为重言式。**
* **⑶如果每个真值赋值都不满足Q，则称Q为永假式，也称为矛盾式，不可满足式。**
* **⑷如果至少有一个真值赋值满足Q，则称Q为可满足式。**

**逻辑等价**

* **定义1.9 设 *A*, *B* 是公式。如果对于每个真值赋值*v*，**
* ***v*(*A*) = *v*(*B*)，**
* **则称 *A* 和 *B* 等值，也称 *A* 与 *B “* 逻辑等价”，**
* **记为 *A*** ⇔ ***B* 。**
* ***A*** ⇔ ***B* 的意思是： *A* 和 *B* 的逻辑语义相同，即**
* **不论公式中的命题变元取什么赋值， 公式*A* 和 *B* 的逻辑真值都是一样的，要真都真，要假都加。**

**等值式模式**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **交换律** | **Q∨R⇔Q∨R** | **Q∧R⇔R∧Q** | **Q⊕R⇔R⊕Q** | |
| **结合律** | **(P∨Q)∨R**  **⇔P∨(Q∨R)** | **(P∧Q)∧R**  **⇔P∧(Q∧R)** | **(P⊕Q)⊕R**  **⇔P⊕(Q⊕R)** | |
| **分配律** | **P∨(Q∧R)**  **⇔(P∨Q)∧(P∨R)** | **P∧(Q∨R)**  **⇔(P∧Q)∨(P∧R)** | **P∧(Q⊕R)**  **⇔(P∧Q)⊕(P∧R)** | |
| **德•摩根律** | **¬(Q∨R)⇔¬Q∧¬R** | **¬(Q∧R)⇔¬Q∨¬R** | | |
| **幂等律** | **Q∨Q⇔Q** | **Q∧Q⇔Q** | | |
| **同一律** | **Q∧1⇔Q** | **Q∨0⇔Q** | | |
| **吸收律** | **Q∨(Q∧R)⇔Q** | **Q∧(Q∨R)⇔Q** | | |
| **零 律** | **Q∨1⇔1** | **Q∧0⇔0** | | |
| **排中律** | **Q∨¬Q⇔1** | **双重否定律** | | **¬ ¬Q⇔Q** |
| **矛盾律** | **Q∧¬ Q⇔0** | **假言易位** | | **Q→R⇔ ¬ R→¬ Q** |

**对偶定理**

* **定义1.10 设 *A* 是由 { 0, 1,** ¬**,** ∨**,** ∧**} 生成的公式，**
* **将 *A* 中的** ∨**和** ∧**互换，0和 1互换得到 *A*\* ，**
* **称 *A*\* 与 *A* 互为对偶式。**
* **比如： (*p*** ∨ ***q*)** ∧ ***r* 与 (*p*** ∧ ***q*)** ∨ ***r* 互为对偶式，**
* ¬ **(*p*** ∨ **0)** ∧ **1与** ¬ **(*p*** ∧ **1)** ∨ **0 互为对偶式。**
* **定义1.11 如果真值赋值 *v*1 和 *v*2 满足：**
* **对于每个命题变元 *p* ， *v*1(*p*)** ≠ ***v*2(*p*) ，**
* **则称 *v*1 和 *v*2 是相反的赋值。**
* **相反赋值的含义理解为：*v*1(*p*) =** ¬ ***v*2( *p*) = *v*2(**¬ ***p*)。**
* **若 *v*1(*A*) 已知，*v*2(*A*) 是对*v*1(*A*) 所有命题变元取相反赋值**
* **定理1.4 设 *A* 是由 {0, 1,** ¬**,** ∨**,** ∧**} 生成的公式，**
* ***A*\*与 *A* 互为对偶式，*v* 和 *v***′ **是相反的，则**
* ***v*(*A*\*) =** ¬***v***′**(*A*)**

**完全集**

* **设S是联结词集合。如果每个n(n>0)元的联结词都可由S定义，则称S为完全集。**
* **定理⒈６ {**¬**,** ∧**,** ∨**}是完全集。**

**命题逻辑的公理系统**

* **弗雷格公理系统**
  + Q→(R→Q)
  + (P→(Q→R)) →((P→Q) →(P→R))
  + (P→(Q→R)) →(Q→(P→R))
  + (Q→R) →(¬R→¬Q)
  + ¬¬Q→Q
  + Q→¬¬Q
* **卢卡西维茨公理系统**
  + Q→(R→Q)
  + (P→(Q→R)) →((P→Q) →(P→R))
  + **(**¬Q→¬R) →(R →Q)
* **罗素公理系统**
  + Q∨Q →Q
  + Q→Q∨R
  + Q∨R→R∨Q
  + (P→Q)→(P∨R→Q∨R)
* **可靠性：公理系统中的每条定理都是永真式**
* **完备性 ：通过公理系统能推出一切永真式，即每个**

**永真式都是公理系统的定理。**

* **独立性：如果一个公理系统没有多余的公理模式和**

**推理规则。即，去掉任何一个公理模式或**

**推理规则，定理都会减少。**

* + **公理：公理模式中A, B, C为任意公式**

**公理模式A*1*： A→ (B→A)**

**公理模式A2： (A→ (B→C)) → ((A→B) → (A→C))**

**公理模式A3： (¬A→¬B) → (B→A)**

* + **推理规则：(分离规则，MP(Modus Ponents)规则)**

**若A和A→B成立，则B成立。其中， A和A→B称为前提**

**B称为结论。**

* + **定理集**
  + **谓词公理系统中仅使用了**¬**和**→**联结词符号，而其他联结词符号**∨**,** ∧**,** ↔**,** ⊕**可以认为是缩写公式，用≡表示缩写定义。**

**(1) A∨B ≡ (¬A→B)**

**(2) A∧B ≡ ¬ (A→¬B)**

**(3) A↔B ≡ (A→B) ∧(B→A)**

**(4) A⊕B ≡ ¬ (A↔B)**

**谓词逻辑公理系统**

* **谓词逻辑的公理系统定义：**

**(1) 符号集合：**

* + **个体变元：x1, x2, …**
  + **个体常元：c1, c2 , …**
  + **函词符号：f11, f21,......；f12, f22,......；**
  + **谓词符号：Q11,Q21,......；Q12, Q22,....;**
  + **运算符号：**∀**,** ¬**,** →**；**
  + **逗 号：, ;**
  + **括 号：(, )**

**(2) 项定义：**

* + **个体常元是项；**
  + **个体变元是项；**
  + **若是t1,…,tn项，则是fkn (t1,…,tn)项。**

**(3) 公式集合：**

* + **若是t1,…,tn项，则A kn (t1,…,tn)是公式。**
  + **若A是公式，则(**¬**A)是公式；**
  + **若A和B是公式，则(A**→**B)是公式；**
  + **若A是公式，则(**∀**xA)是公式。**

**(4) 公理集合：**

* + **公理模式A 1：A**→ **(B**→**A)**
  + **公理模式A 2：(A**→ **(B**→**C))** → **((A**→**B)** →**(A**→**C))**
  + **公理模式A 3：(**¬**A**→¬**B)** → **(B**→**A)**
  + **公理模式A 4：**∀**xA(x)**→**A(x)[x/t]，其中项t对于A中的x是可代入的。**
  + **公理模式A 5：**∀**x(A**→**B(x))** → **(A**→∀**xB(x))，其中x不是A中自由变元。**

**(5) 推理规则**

* + **分离规则（简称MP规则）：从A和A**→**B推出B。**
  + **概括规则（简称UG规则）：从A推出(**∀**xA)。**