Discrete Mathematics - P_1

命题 (proposition)是一个陈述语句，或真或假，但不能即真又假(即二态的 True/False)

命题变元 (proposition variable)是代表命题的变量，其真值(truth value)表示命题为真或假(T/F)

命题演算 或称命题逻辑，是涉及命题的逻辑领域，由亚里士多德(Aristotle)系统地创建

$\neg p$ ($\bar{p}$) (p的否定，negation of p，也称非p) 指"不是p所指的情形"，其真值与p相反

复合命题 (compound proposition)，是用逻辑运算符(logical operator)组合命题所构造出的命题

真值表 (truth table) 显示命题所有可能真值的表，复合命题总是有相同真值时，称为等价的(equivalent)
(真值表相同)

p	T(1)	T(1)	F(0)	F(0)	
q	T(1)	F(0)	T(1)	F(0)	
$p\wedge q$	1	0	0	0	合取(conjunction)，"p与q"，当且仅当p,q为真时 $p\wedge q$为真 (and)
$p\vee q$	1	1	1	0	析取(disjunction)，"p或q"，当且仅当p,q为假时 $p\vee q$为假(or)
$p\oplus q$	0	1	1	0	异或(exclusive or)，"p XOR q"，相对(or)也叫兼或(inclusive or)
$p\to q$	1	0	1	1	蕴含(p implies q)，"如果p，则q"，p为假设(前提)(前件)，q为结论(后件)
$q\to p$	1	1	0	1	逆命题(converse of $p\to q$)，互为逆否命题 特别注意：常见的逻辑错误为假设(认定)
$\neg p\to\neg q$	1	1	0	1	(否)反命题(inverse of $p\to q$) 逆命题/反命题与原命题等价
$\neg q\to\neg p$	1	0	1	1	逆否命题(contrapositive of $p\to q$)，条件语句与逆否命题等价
$p\leftrightarrow q$	1	0	0	1	当且仅当(if and only if)，也称双条件(biconditional)
$p\mid q$	0	1	1	1	与非(NAND)，"p NAND q"，(| 称为竖线 Sheffer)
$p\downarrow q$	0	0	0	1	或非(NOR)，"p NOR q"，(↓称为箭头 Peirce)
$\neg p$	0	0	1	1	非(NOT)，"非p"

优先级 $\neg$(否定) > $\wedge$(合取) > $\vee$(析取) > $\to$(条件) > $\leftrightarrow$(双条件)

布尔变量 (Boolean variable)，用1/0代替T/F，用于计算机位运算(bit operation) 如：AND, OR, NOT, XOR, NAND, NOR

按位运算 (bitwise operation)，定义在位串(bit string)上的运算，对两个位串的对应位进行位运算

逻辑门 (logic gate)，对一个或多个位执行逻辑运算以产生输出位的逻辑单元

逻辑电路 (logic circuit)，由逻辑门构成，产生一个或多个输出位的开关电路

Discrete Mathematics - P_2

模糊逻辑，即命题的真值为$[0,1]$的一个数，以表示真实性的不同程度，用于人工智能等领域
定义运算：否定为$1-$原命题真值，合取为若干命题真值的最小值，析取为若干命题真值的最大值

规范描述 (specification)，将数学，自然科学，自然语言中的语句翻译成逻辑语言，精确表达以避免歧义
系统规范说明应该是一致的，不应包含可能导致矛盾的相互冲突的需求
即对于规范说明中的所有命题，存在一组命题真值，使得所有规范描述的逻辑语句均为真

布尔搜索 在信息搜索中运用逻辑连接词(如AND, OR, NOT)和命题逻辑的技术

逻辑迷题 可以用逻辑推理解决的迷题。求解迷题是实践逻辑规则的方法
泥巴孩子迷题(muddy children puzzle)，类似红眼问题，此类问题涉及共有知识与公共知识
共有知识(mutual knowledge)，所有人都知道事件发生，但没有给出是否知道其他人知道的信息
公共知识(common knowledge)，P是common knowledge $\to$ P是mutual knowledge，反之不成立
即所有人都知道P，所有人都知道所有人都知道P，all know that all know that all know P … ad infinitum

门电路 (gate)，非门(NOT，逆变器) $p \to \triangleright\!\circ \to \neg p$，或门(OR) $p, q \to p\vee q$，与门(AND) $p, q \to p\wedge q$

永真式/重言式 (tautology)永远为真(T)的命题，永远为假(F)的命题为矛盾式(contradiction)，其余(T/F)为可能式(contingency)

逻辑等价 (logically equivalent)，对于任何命题变元的真值组合，都有相同真值的命题，即有相同的真值表 (方法1)
定义为如果$p\leftrightarrow q$是永真式(T)，则p与q是逻辑等价的，用$p\equiv q$表示(也写作$p\Leftrightarrow q$)
判断逻辑等价式的另一种方法是，存在$r_1,\dots,r_n$命题，使得$p\equiv r_1\equiv\cdots\equiv r_n\equiv q$成立，则$p\equiv q$

逻辑等价式 logically equivalent compound proposition

- 恒等律：$p\wedge T\equiv p$，$p\vee F\equiv p$
- 支配律：$p\vee T\equiv T$，$p\wedge F\equiv F$
- 幂等律：$p\vee p\equiv p$，$p\wedge p\equiv p$
- 双重否定律：$\neg(\neg p)\equiv p$
- 交换律：$p\vee q\equiv q\vee p$，$p\wedge q\equiv q\wedge p$
- 结合律：$(p\vee q)\vee r\equiv p\vee(q\vee r)$，$(p\wedge q)\wedge r\equiv p\wedge(q\wedge r)$
- 分配律：$p\vee(q\wedge r)\equiv(p\vee q)\wedge(p\vee r)$，$p\wedge(q\vee r)\equiv(p\wedge q)\vee(p\wedge r)$
- (De Morgan) 德·摩根律：$\neg(p\wedge q)\equiv\neg p\vee\neg q$，$\neg(p\vee q)\equiv\neg p\wedge\neg q$
- 吸收律：$p\vee(p\wedge q)\equiv p$，$p\wedge(p\vee q)\equiv p$
- 否定律：$p\vee\neg p\equiv T$，$p\wedge\neg p\equiv F$

- 条件命题：$p\to q\equiv\neg p\vee q$，$\neg(p\to q)\equiv p\wedge\neg q$
- 逆否命题：$p\to q\equiv\neg q\to\neg p$
- 析取/合取：$p\vee q\equiv\neg p\to q$，$p\wedge q\equiv\neg(p\to\neg q)$
- 条件分配：$(p\to q)\vee(p\to r)\equiv p\to(q\vee r)$，$(p\to q)\wedge(p\to r)\equiv p\to(q\wedge r)$，$(p\to r)\wedge(q\to r)\equiv(p\vee q)\to r$，$(p\to r)\vee(q\to r)\equiv(p\wedge q)\to r$
- 双条件定义：$p\leftrightarrow q\equiv(p\to q)\wedge(q\to p)$ (充分必要条件)
- 双条件否定：$p\leftrightarrow q\equiv\neg p\leftrightarrow\neg q$ (等价两侧取反等价)
- 析取范式：$p\leftrightarrow q\equiv(p\wedge q)\vee(\neg p\wedge\neg q)$
- 否定双条件：$\neg(p\leftrightarrow q)\equiv p\leftrightarrow\neg q$，$\neg(p\leftrightarrow q)\equiv p\oplus q$

Discrete Mathematics - P3

扩展德·摩根律 德·摩根律可以扩展至n个命题析取/合取的形式 $\neg(\bigvee_{i=1}^{n} p_i)=\bigwedge_{i=1}^{n}\neg p_i$, $\neg(\bigwedge_{i=1}^{n} p_i)=\bigvee_{i=1}^{n}\neg p_i$

可满足的复合命题 (satisfiable compound proposition), 存在一组对命题变元的真值赋值, 使命题为真, 矛盾式是不可满足的(当且仅当)

这组赋值称为特定的可满足性问题的一个解, 可以通过真值表或析取范式得到

相容的复合命题 (consistent compound propositions), 存在一组对命题变元的真值赋值, 使一组命题全部为真

如数独问题可以写成三个复合[redacted]命题[redacted] 每一行包含每一个数 $\bigwedge_{i=1}^{9}\bigwedge_{n=1}^{9}\bigvee_{j=1}^{9} P(i,j,n)$

有唯一一组命题变元赋值使其为真 每一列包含每一个数 $\bigwedge_{j=1}^{9}\bigwedge_{n=1}^{9}\bigvee_{i=1}^{9} P(i,j,n)$

令P(i,j,n)为命题第i行,第j列包含数字n 每一个九宫格包含每一个数: $\bigwedge_{r=0}^{2}\bigwedge_{s=0}^{2}\bigwedge_{n=1}^{9}\bigvee_{i=1}^{3}\bigvee_{j=1}^{3} P(3r+i,3s+j,n)$

对偶式 (p^*) 为P[redacted]中每个∨改为∧, 每个∧改为∨, 每个T改为F, 每个F改为T所形成的命题, 记为 (p^*)

如果对P中每个变元 p_i 以 $\neg p_i$ 替代, 新命题等价与 $\neg P$, 则 $P\equiv p^*$ ($P(\neg p_1,\dots,\neg p_n)\equiv\neg P^*(p_1,\dots,p_n)$)

对偶的对偶与原命题等价, 即 $(p^*)^*\equiv p$

对偶定理 对只含运算符 ∨, ∧, ¬ 的复合命题, 如果 $p\equiv q$, 则 $p^*\equiv q^*$

析取范式 对于任何复合命题 $P(p_1,\dots,p_n)$, 可写为由 $(p_1/\neg p_1),\dots,(p_n/\neg p_n)$ 的合取式的析取式, 每个合取式代表一组使命题 $P(p_1,\dots,p_n)$ 为真的变元赋值组合, 即 $P(p_1,\dots,p_n)=\bigvee_{j=0}^{m}(\bigwedge_{i=1}^{n} y_{ij})$, 其中 y_{ij} 取 p_i 或 $\neg p_i$, $m\le 2^n$

由此引出n个变元的命题P共有2^2^n种不同可能

功能完备集 一组逻辑[redacted]运算符, 如果所有复合命题都等价于一个只含有这些逻辑运算符的复合命题.

{¬, ∨, ∧} 是功能完备集, 因为析取范式 $P(p_1,\dots,p_n)=\bigvee_{j=0}^{m}(\bigwedge_{i=1}^{n}(p_i\text{或}\neg p_i))$

{¬, ∨} [redacted] {¬, ∧} 是功能完备集, 因为 $p\wedge q\equiv\neg(\neg p\vee\neg q)$; $p\vee q\equiv\neg(\neg p\wedge\neg q)$

{|} ([redacted]与非, NAND) 是功能完备集, 因为 $\neg p\equiv p|p$; $p\wedge q\equiv(p|q)|(p|q)$, $p\vee q\equiv(p|p)|(q|q)$

{↓} (或非, NOR) 是功能完备集, 因为 $\neg p\equiv p\downarrow p$, $p\vee q\equiv(p\downarrow q)\downarrow(p\downarrow q)$, $p\wedge q\equiv(p\downarrow p)\downarrow(q\downarrow q)$

谓词 (predicate) 句子中代表主语属性 (property) 的部分, 是一种表达能力更强的逻辑

命题函数 (propositional function) 包含变量的语句, [redacted]当每一个变量被赋值或被量词约束时, 即成为命题

即形如 $P(x_1,\dots,x_n)$ 的命题函数, 称为n位谓词或n元谓词, 其中变量 $x_1,\dots,x_n$ 可以是任何类型

[redacted]

谓词可以用来验证计算机程序, 即证明当给定合法输入时, 计算机程序总是能产生期望的输出

注意: 除非建立程序的正确性, 或验证了每个输入值, 否则无论测试了多少组都无法证明对所有输入有效

前置条件 描述合法输入应满足条件的语句 (pre-condition)

后置条件 描述程序运行的输出应该满足的条件 (post-condition) } 构成基本的程序完整性检查

Discrete Mathematics - P4

量化 表示在何种程度上谓词对于一定范围的个体成立，从命题函数生成一个命题，如自然语言(所有，某些，许多，少量，没有)
处理谓词和量词的逻辑领域称为谓词演算

论域 (domain of discourse)，或称全体域(universe of discourse)，或简称域(domain)
使用量词时必须指定论域，否则量化是无意义的，论域改变时量化的意义随之改变

全称量化 语句"$P(x)$对x在其论域的所有值为真"，用符号 $\forall_x P(x)$ 表示，$\forall$为全称量词
$\forall_x P(x)$ 读作"对所有/每个x，$P(x)$"，使$P(x)$为假的个体称为$\forall_x P(x)$的反例
特别注意：如果论域为空，$\forall_x P(x)$对任何$P(x)$都为真，因为不存在一个论域中的x使$P(x)$为假
另外注意：尽量避免使用"对任一x"，因为可能引起歧义，如"每个"或"某些"
$\forall_x P(x)$ 等价于所有论域内命题的合取，即有 $\forall_x P(x) \equiv \bigwedge_{x\in D} P(x)$

存在量化 语句"论域中存在一个个体x使$P(x)$为真"，用符号 $\exists_x P(x)$ 表示，$\exists$为存在量词
$\exists_x P(x)$ 读作"至少有一个/对某些x 使$P(x)$为真"
特别注意：如果论域为空，$\exists_x P(x)$对任何$P(x)$都为假，因为不存在一个论域中的x使$P(x)$为真
$\exists_x P(x)$ 等价于所有论域内命题的析取，即有 $\exists_x P(x) \equiv \bigvee_{x\in D} P(x)$

唯一性量词 用$\exists!$或$\exists_1$表示，$\exists!_x P(x)$ / $\exists_{1x} P(x)$ 表示"存在一个唯一的/有且仅有一个x使$P(x)$为真"

$$\exists!_x P(x) \equiv P(x_0) \land \forall_{x\in D}(x\neq x_0) \to \neg P(x) \equiv \forall_{x\in D}(x\neq x_0) \leftrightarrow \neg P(x_0)$$

命题	$\forall_x P(x)$	$\exists_x P(x)$	$\exists!_x P(x)$
为真	对每一个x，$P(x)$都为真	有一个x，使$P(x)$为真	有且仅有一个x，使$P(x)$为真
为假	有一个x，使$P(x)$为假	对每一个x，$P(x)$都为假	有至少2个x，使$P(x)$为真，或对每一个x，$P(x)$都为假

量词的作用域 (scope of a quantifier) 语句中量词作用的部分，命题函数所有变量必须是约束的或者设置为等于某值的，才能转变为一个命题，可以通过全称量词，存在量词和赋值实现
有量词作用变量x时，称为约束变量(bound variable)，即被量化的变量
在所有限定变量的量词的作用域之外的变量称为自由变量(free variable)，即命题中未绑定的变量

量词谓词逻辑等价 当且仅当无论代入什么谓词，无论为命题函数的变量指定什么论域，都有相同真值，用$S\equiv T$表示

包括合取与析取

量词分配 全称量词对于合取式是可分配的 $\forall_x(P(x)\land Q(x)) \equiv \forall_x P(x) \land \forall_x Q(x)$
存在量词对于析取式是可分配的 $\exists_x(P(x)\lor Q(x)) \equiv \exists_x P(x) \lor \exists_x Q(x)$
注意这里要求$P(x)$与$Q(x)$有相同的论域

Discrete Mathematics - P5

量词分配 全称量词对析取式不可分配，如有 x_1 使 $P(x_1)$ 为真，$Q(x_1)$ 为假；x_2 使 $P(x_2)$ 为假，$Q(x_2)$ 为真，则 $\forall x(P(x)\vee Q(x))\equiv T$，$\forall x P(x)\equiv F$，$\forall x Q(x)\equiv F$

存在量词对合取式不可分配，如有 $(x_1\neq x_2)$ x_1 使 $P(x_1)$ 为真，$Q(x_1)$ 为假；x_2 使 $P(x_2)$ 为假，$Q(x_2)$ 为真，则 $\exists x P(x)\equiv T$，$\exists x Q(x)\equiv T$，$\exists x(P(x)\wedge Q(x))\equiv F$

量词否定 / 德·摩根律

$\neg\forall x P(x)\equiv\neg\bigwedge_{x\in D}P(x)\equiv\bigvee_{x\in D}\neg P(x)\equiv\exists x\neg P(x)$，

$\neg\exists x P(x)\equiv\neg\bigvee_{x\in D}P(x)\equiv\bigwedge_{x\in D}\neg P(x)\equiv\forall x\neg P(x)$

注意到推导过程使用了扩展德摩根律

约束论域 限定量词的论域会采用简写，即将变量必须满足的条件直接放在量词后面，如 $\forall x\in D\ P(x)$，$\exists x\in D\ P(x)$

但特别注意将简写表示写成完整命题时全称量词与存在量词是不同的

复合命题形式 ← $\forall_{x\in D}P(x)\equiv\forall x((x\in D)\to P(x))$，$\exists_{x\in D}P(x)\equiv\exists x((x\in D)\wedge P(x))$

注意对于全称量词，无需 $(x\in D)\wedge P(x)$ 是因为不对 $x\notin D$ 的部分的真值做要求，

而对于存在量词，$(x\in D)\to P(x)$ 不足是因为所有 $(x\notin D)$ 的部分皆使 $(x\in D)\to P(x)$ 为真，无从保证 $P(x)$ 在 $(x\in D)$ 中为真

Prolog (Programming in Logic) 使用谓词逻辑的规则进行推理的程序设计语言

Prolog 事实通过指定满足谓词的元素来定义谓词

Prolog 规则使用 Prolog 事实定义的谓词来定义新的谓词　　查询/返回

Prolog 事实 如：

```
instructor(chan, math273)          ?instructor(patel, ee222) → yes
instructor(patel, ee222)           ?enrolled(X, ee222) → juana
enrolled(juana, ee222)             ?teaches(X, juana) → patel
```

Prolog 规则

```
teaches(P, S) :- instructor(P, C), enrolled(S, C)
```

注意：Prolog 中逗号用于表示谓词合取，分号用于谓词析取

嵌套量词 对于嵌套的量化式，可以借住嵌套循环（即将量化当作循环）来思考，即类似于程序中循环嵌套

即对如 $\forall x P(x)$ 可以用形式为 `for each x in D { if not P(x) { return False; } } return True;`

$\exists x P(x)$ 可以写为 `for each x in D { if P(x) { return True; } } return False;`

量词顺序 注意嵌套量词的顺序不同可能形成不同的命题逻辑，所在需要特别注意嵌套的顺序

语句	$\forall x\forall y P(x,y)$ $\forall y\forall x P(x,y)$ →	$\exists x\forall y P(x,y)$ →	$\forall y\exists x P(x,y)$ →	$\exists x\exists y P(x,y)$ $\exists y\exists x P(x,y)$	注意从左向右，依次存在蕴含关系（充分条件）
为真	对每一对 (x,y) $P(x,y)$ 均为真	存在一个 x，使对每一个 y $P(x,y)$ 均为真	对每一个 y 存在一个 x 使 $P(x,y)$ 为真	存在一对 (x,y) 使 $P(x,y)$ 为真	从右向左依次为必要条件
为假	存在一对 (x,y) 使 $P(x,y)$ 为假	对每一个 x，存在一个 y 使 $P(x,y)$ 为假	存在一个 y 使对每一个 x $P(x,y)$ 均为假	对每一对 (x,y) $P(x,y)$ 均为假	
合取/析取	$\bigwedge_x\bigwedge_y P(x,y)$ 等价于 $\forall(x,y)P(x,y)$	$\bigvee_x\bigwedge_y P(x,y)$	$\bigwedge_y\bigvee_x P(x,y)$	$\bigvee_y\bigvee_x P(x,y)$ 等价于 $\exists(x,y)P(x,y)$	

Discrete Mathematics - P6

[redacted] 假设A为一个命题语句，且x在A中不作为自由变量出现，且假设[redacted]论域非空

空量化 null quantification

$\forall x\, P(x) \vee A \equiv (\bigwedge_x P(x)) \vee A \equiv \bigwedge_x (P(x) \vee A) \equiv \forall x\,(P(x) \vee A)$ （分配律）

$\exists x\, P(x) \vee A \equiv (\bigvee_x P(x)) \vee A \equiv \bigvee_x (P(x) \vee A) \equiv \exists x\,(P(x) \vee A)$ （[redacted]幂等律，结合律）

$\forall x\, P(x) \wedge A \equiv (\bigwedge_x P(x)) \wedge A \equiv \bigwedge_x (P(x) \wedge A) \equiv \forall x\,(P(x) \wedge A)$ （幂等律，结合律）

$\exists x\, P(x) \wedge A \equiv (\bigvee_x P(x)) \wedge A \equiv \bigvee_x (P(x) \wedge A) \equiv \exists x\,(P(x) \wedge A)$ （分配律）

$\forall x\,(P(x) \rightarrow A) \equiv \bigwedge_x (\neg P(x) \vee A) \equiv (\bigwedge_x \neg P(x)) \vee A \equiv \neg(\bigvee_x P(x)) \vee A \equiv \exists x\, P(x) \rightarrow A$ （分配律，德摩根律）

$\exists x\,(P(x) \rightarrow A) \equiv \bigvee_x (\neg P(x) \vee A) \equiv (\bigvee_x \neg P(x)) \vee A \equiv \neg(\bigwedge_x P(x)) \vee A \equiv \forall x\, P(x) \rightarrow A$ （分配律，德摩根律）

$\forall x\,(A \rightarrow P(x)) \equiv \bigwedge_x (\neg A \vee P(x)) \equiv \neg A \vee (\bigwedge_x P(x)) \equiv A \rightarrow \forall x\, P(x)$ （分配律）

$\exists x\,(A \rightarrow P(x)) \equiv \bigvee_x (\neg A \vee P(x)) \equiv \neg A \vee (\bigvee_x P(x)) \equiv A \rightarrow \exists x\, P(x)$ （幂等律，结合律）

量词优先级 量词$\forall$，$\exists$和$\exists!$ ($\exists_1$) 具有[redacted]比所有命题演算的逻辑运算符更高的优先级

嵌套量词否定 可以通过连续地应用单个量词语句的否定规则得到，如 $\neg \forall x \exists y\, P(x,y) \equiv \exists x (\neg \exists y\, P(x,y)) \equiv \exists x \forall y \neg P(x,y)$

前束范式 (prenex normal form, PNF)，即语句为形如 $Q_1 x_1 Q_2 x_2 \cdots Q_k x_k P(x_1, \ldots, x_k)$

其中 Q_i $(i=1,2,\cdots,k)$ 为全称量词$\forall$或存在量词$\exists$，且 $P(x_1,\cdots,x_n)$ 是不含量词的谓词

即如 $\exists x \forall y (P(x,y) \wedge Q(y))$ 是前束范式，而 $\exists x P(x) \vee \forall x Q(x)$ 则不是

特别注意：每个由命题变量、谓词、T/F，并用逻辑连接词和量词构成的语句都等价于一个前束范式

论证 (argument) 指一连串的命题并以结论为最后的命题，数学上的证明是建立数学命题真实性的有效论证

有效性 (valid) 指结论(conclusion)或论证的最后一个命题必须根据前提(premise)或论证过程前面的命题的真实性推出

即一个论证是有效的当且仅当不可能出现所有前[redacted]提(premise)为真而结论(conclusion)为假的情况

或者说一个论证是有效的，如果它的所有前提为真蕴含着结论为真，即 $\bigwedge_i p_i \rightarrow q \equiv T$

论证形式 (argument form) 是一连串涉及命题变元的复合命题

有效论证形式 (valid argument form)，一连串包含命题变量的复合命题，其中所有前提为真蕴含着结论为真

无论用什么特定命题来替换其中的命题变元，如果前提均真时结论为真，则论证形式是有效的

有效论证 (valid argument)，当 $\bigwedge_{i=1}^{n} p_i \rightarrow q$ 为永真式时，有前提为 $p_1,\cdots,p_n$ 及结论为q的论证形式是有效的

可以先建立一些相对简单的论证形式(称为推理规则)的有效性

这些推理规则可以作为基本构件用来构造更为复杂的有效论证形式

Discrete Mathematics - P7

推理规则							
$p\to q$	$p\to q$	$p\to q$	$p\lor q$	[redacted]		p	$p\lor q$
p	[redacted] $\neg q$	$q\to r$	$\neg p$	p	$p\land q$	q	$\neg p\lor r$
$\therefore q$	$\therefore \neg p$	$\therefore p\to r$	$\therefore q$	$\therefore (p\lor q)$	$\therefore p$	$\therefore p\land q$	$\therefore q\lor r$
永真式 $(p\land(p\to q))\to q$	$(\neg q\land(p\to q))\to\neg p$	$((p\to q)\land(q\to r))\to r$	$((p\lor q)\land\neg p)\to q$	$p\to(p\lor q)$	$(p\land q)\to p$	$((p)\land(q))\to(p\land q)$	$((p\lor q)\land(\neg p\lor r))\to(q\lor r)$
假言推理	取拒式	假言三段论	析取三段论	附加律	化简律	合取律	消解律

(符号∴表示"所以")

假言推理 (modus ponens), 或称分离规则 (law of detachment), 基础是永真式 $(p\land(p\to q))\to q$

拉丁文 modus ponens 意为确认模式 (mode [redacted] that affirms)

消解律 (resolution), 用于在计算机程序中将[redacted]定理的推理和证明任务自动化,

基础是永真式 $((p\lor q)\land(\neg p\lor r))\to(q\lor r)$, 其中 $(q\lor r)$ 称为消解式 (resolvent)

令 $q=r$ 时 有 $((p\lor q)\land(\neg p\lor q))\to q \equiv ((p\land\neg p)\lor q)\to q\equiv q\to q\equiv T$

令 $r\equiv F$ 时 有 $((p\lor q)\land(\neg p))\to q$, 即析取三段论

特别注意消解律的等价形式 $((\neg p\to q)\land(p\to r))\to(q\lor r)$

$$\begin{array}{l} \neg p\to q \\ p\to r \\ \hline \therefore q\lor r \end{array}$$

可以解释为当如果非p则q和如果p则r都为真, 则q和r至少有一个为真, 并有推理

消解律[redacted]可以用来构建自动证明系统, [redacted] 子句指[redacted]变量或其否定的析取式

子句 (clause) [redacted] 用以表示假设和结论, 当使用消解律作为仅有的推理规则构造命题逻辑证明

如 $p\lor(q\land r)\equiv(p\lor q)\land(p\lor r)$, $\neg(p\lor q)\equiv(\neg p)\land(\neg q)$, $p\to q\equiv(\neg p\lor q)$, 括号内为子句(右)

谬误 (fallacy) 常被错误地[redacted]当做一个推理规则(甚至一个错误的论证)使用的一种无效论证形式

肯定结论的谬误 (fallacy of affirming the conclusion), 其可能式形式为 $((p\to q)\land q)\to p$

否定假设的谬误 (fallacy of denying the hypothesis), 其可能式形式为 $((p\to q)\land\neg p)\to\neg q$

[redacted]两者都是在p为假且q为真时为假

量化命题	$\forall x P(x)$	$P(c)$, 论域中任意c	$\exists x P(x)$	$P(c)$, 对论域中某个c
推理规则	$\therefore P(c)$, (在论域中	$\therefore \forall x P(x)$	$\therefore P(c)$, 对论域中某个c	$\therefore \exists x P(x)$
	全称实例	全称引入	存在实例	存在引入
	(universal instantiation)	(universal generalization)	(existential instantiation)	(existential generalization)
	从给定前提 $\forall x P(x)$ 为真得出论域中一个特定成员c $P(c)$ 为真的结论	从对论域中所有元素c, $P(c)$ 为真的前提得出 $\forall x P(x)$ 为真的结论	从给定前提 $\exists x P(x)$ 为真得出论域中存在一个元素c使 $P(c)$ 为真的结论	从论域中有一个c使 $P(c)$ 为真的前提推出 $\exists x P(x)$ 为真的结论
			注意: c不能任意选择, 通常仅知道其存在, 但因其存在即可继续论证	注意: 要求c必须在论域中

Discrete Mathematics - P8

全称假言推理(universal modus ponens)，从给定前提 $\forall x(P(x)\rightarrow Q(x))$ 为真和对论域中一个特定元素 a 使 $P(a)$ 为真，推出 $Q(a)$ 为真的结论

$\forall x(P(x)\rightarrow Q(x))$
$P(a)$，论域中一个特定的 a
$\therefore Q(a)$

全称取拒式 (universal modus tollens)，从给定前提 $\forall x(P(x)\rightarrow Q(x))$ 为真和对论域中一个特定元素 a 使 $Q(a)$ 为假，推出 $P(a)$ 为假的结论

$\forall x(P(x)\rightarrow Q(x))$
$\neg Q(a)$，论域中一个特定的 a
$\therefore \neg P(a)$

全称传递性 对具有相同论域的全称量词，从给定前提 $\forall x(P(x)\rightarrow Q(x))$ 为真和 $\forall x(Q(x)\rightarrow R(x))$ 为真，推出 $\forall x(P(x)\rightarrow R(x))$ 为真的结论

$\forall x(P(x)\rightarrow Q(x))$
$\forall x(Q(x)\rightarrow R(x))$
$\therefore \forall x(P(x)\rightarrow R(x))$
注意其中所有的量词均具有相同的论域

证明可以使用定理的假设(如有)，假定为真的公理，以及之前已经被证明的定理

证明 (proof) 对定理为真的展示过程，建立数学语句真实性的有效论证

非形式化证明(informal proof)，指每个步骤会用到多于一条推理规则，会省略步骤，不会显式地列出所用到的假设公理和推理规则

定理 (theorem) 是能够被证明是真的语句，可以证明为真的数学断言，也称事实(fact)或结论(result)
可以是带有若干前提及一个结论的条件语句的全称量化式，或是其他类型的逻辑语句
证明是建立定理真实性的一个有效论证，可以用到公理(axiom)或假设(postulate)

公理 (axiom) 指假设为真的并可作为基础用来证明定理的命题
可以采用无须定义的原始术语陈述，而定理和证明中的所有其他术语都必须是有定义的
术语的定义和推理规则用于从断言推出结论，并绑定在证明的每个步骤

引理 (lemma) 用来证明其他定理的定理，可用一系列引理进行复杂证明，每个引理都可独立被证明

推论 (corollary) 是从一个已经被证明的定理可以直接建立起来的一个定理

猜想 (conjecture) 是被提出认为是真的命题，是真值未知的数学断言
通常基于部分证据，启发式论证或者专家的直觉
注意：很多时候会依据情况而在证明时假定猜想为定理