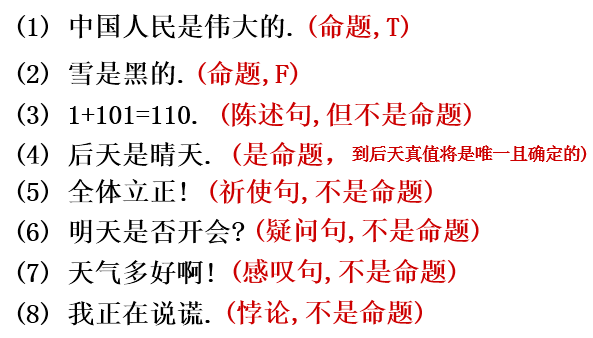
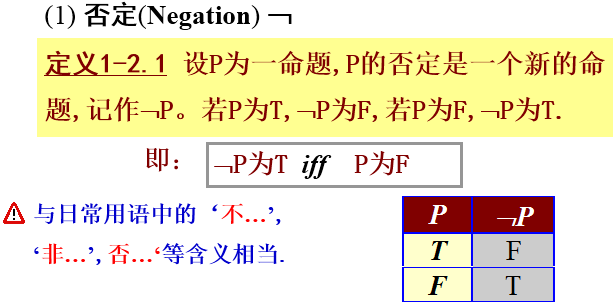
**第一部分 命题逻辑**

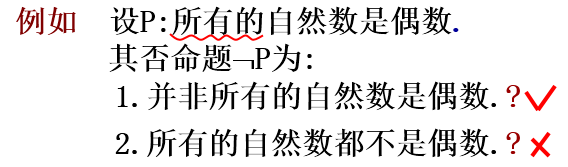
**1. 判断一个句子是不是命题**

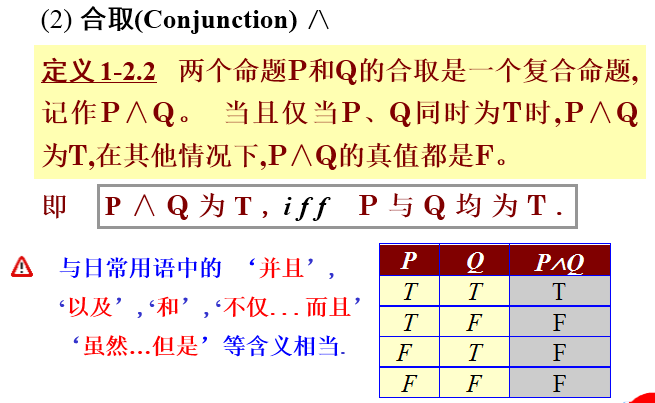
**命题是具有唯一确定真值的陈述句.**



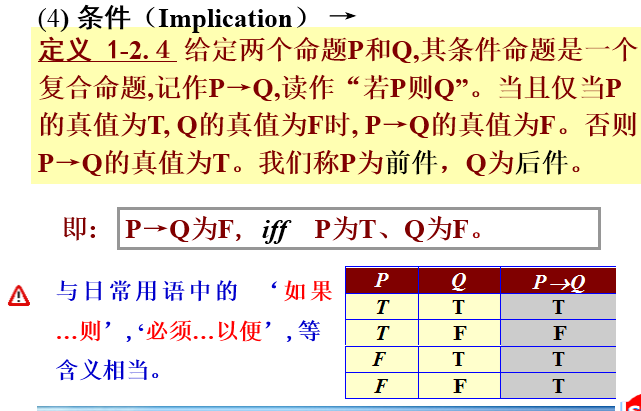
**2. 掌握5种基本逻辑连接词的含义和用法，否定、析取、合取、条件、双条件**

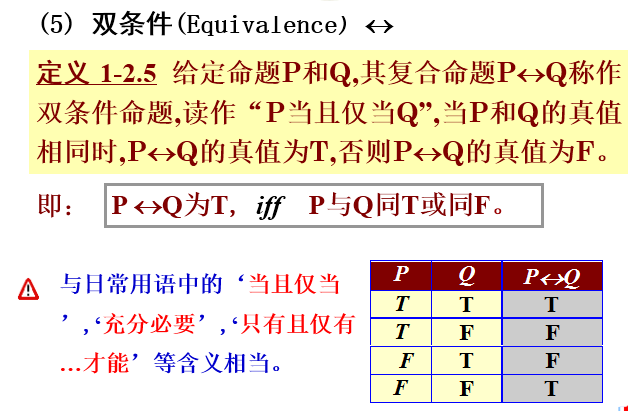










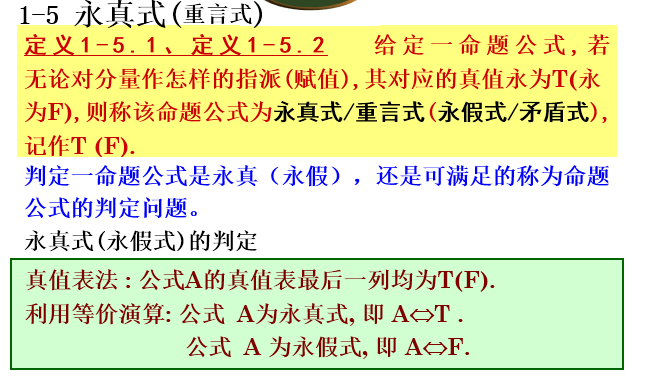


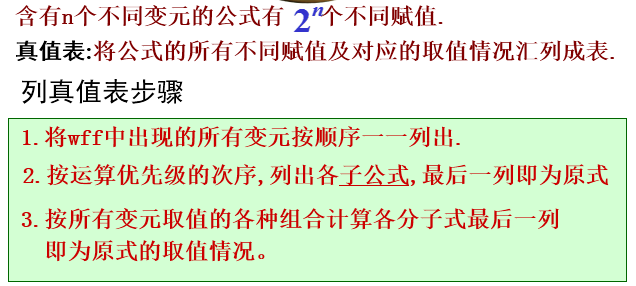
**3. 求命题公式的真值。**

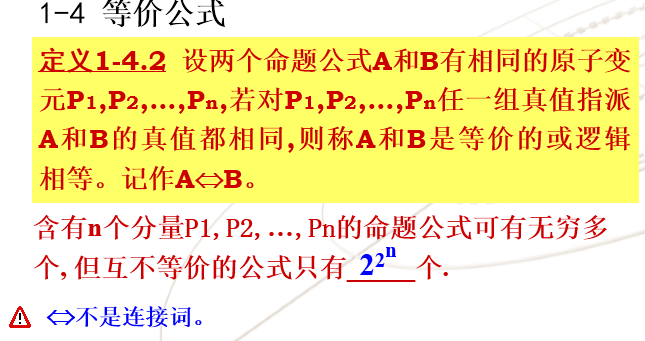
方法：真值表法、等价演算法（重点是等价演算法）

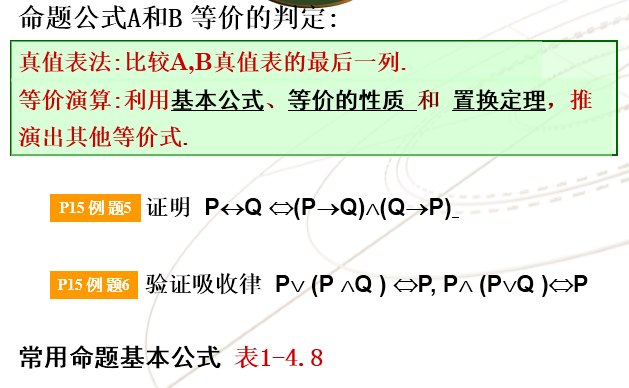
注意：如果命题公式的真值在任何赋值情况下都为真，则为永真式（重言式）；

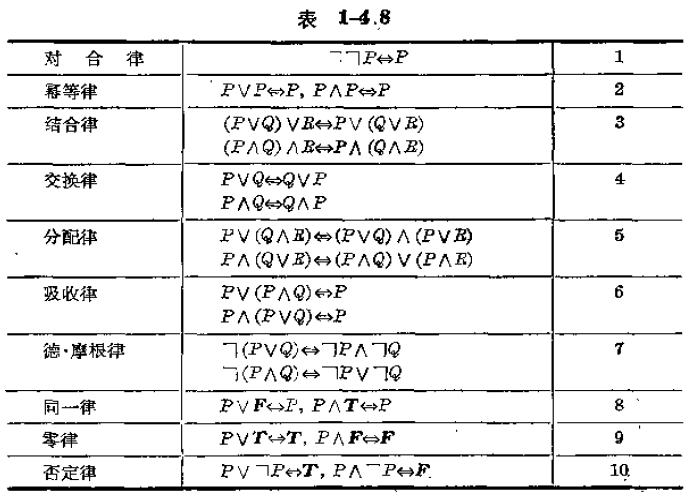
如果命题公式的真值在任何赋值情况下都为假，则为永假式（矛盾式）；

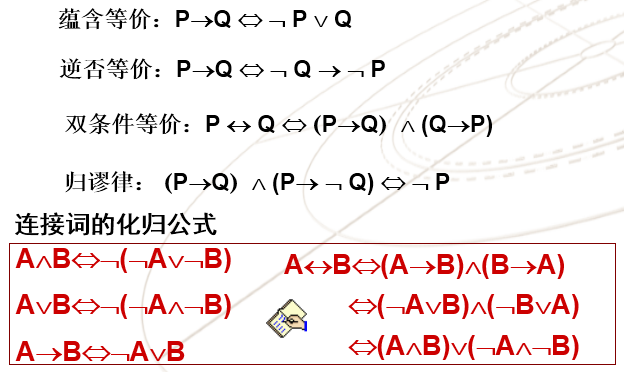


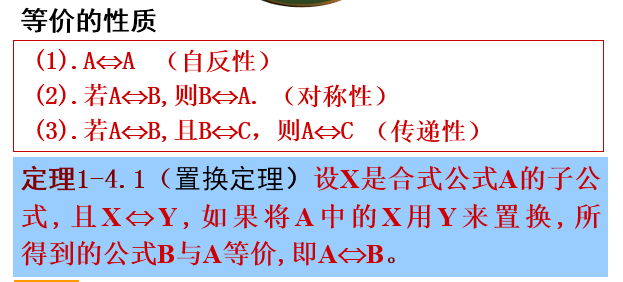




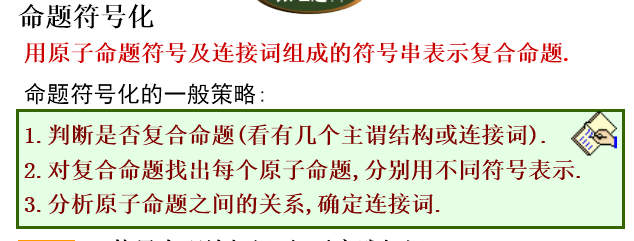


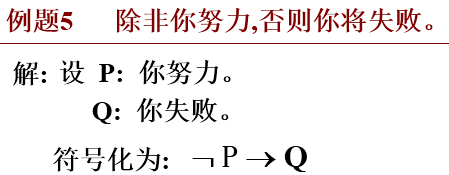






**4. 命题的符号化，给出一个句子，用命题及逻辑连接词表示。**





1. 命题符号化：“除非，否则”

（1）除非你努力，否则你将失败（理解为：“如果你不努力，那么就会失败”）

P：你努力

Q：你失败

⎤P->Q

（2）除非外面不刮风，否则他不在室内打羽毛球（理解为：“如果外面不刮风，那么他就不再室内打羽毛球”）

P：外面不刮风

Q在室内打羽毛球

⎤P->⎤Q

2. 仅当你走，我将留下

P：你走

Q：我留下

Q->P

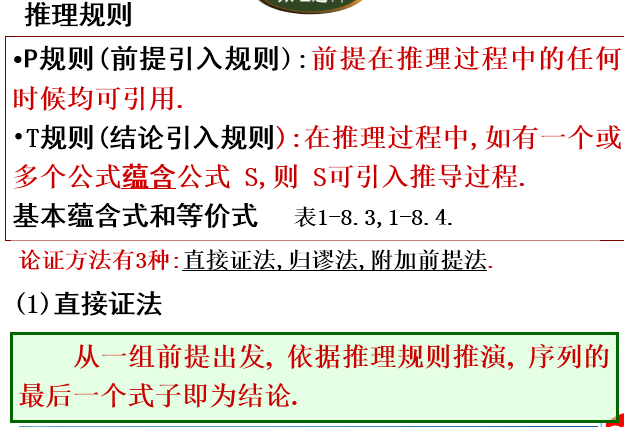
**5. 命题的推理证明**

两种题型：（1）命题公式的推理证明

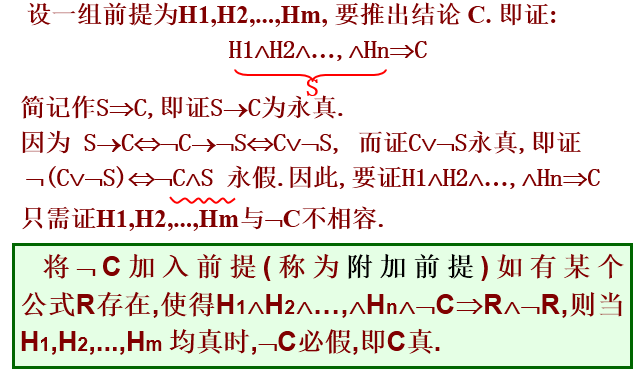
（2）先进行命题符号化，再对符号化得到的命题公式进行推理证明

**考点：**

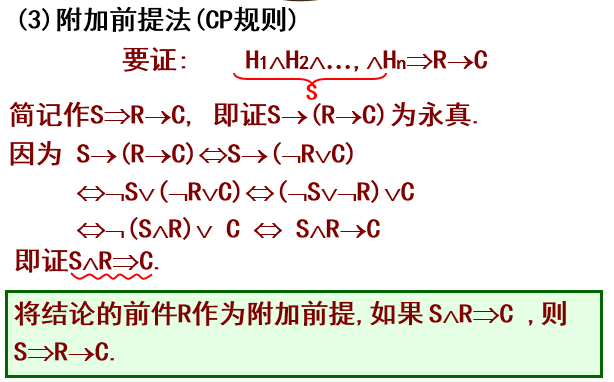
（1）直接证明法



（2）归谬法：在结论前面加⎤连接词，作为附加前提。

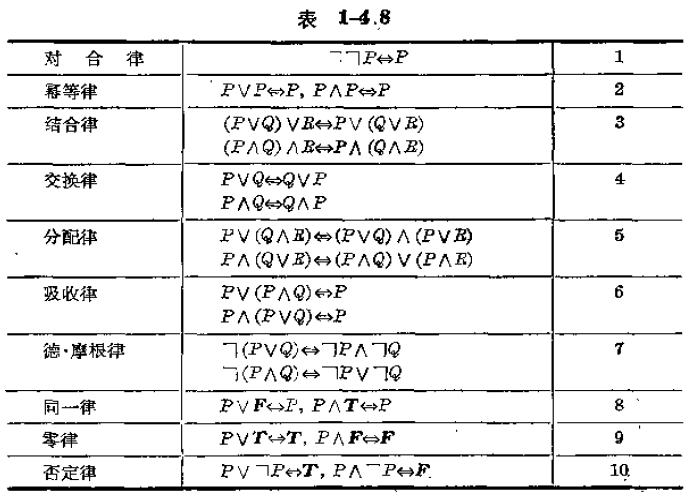


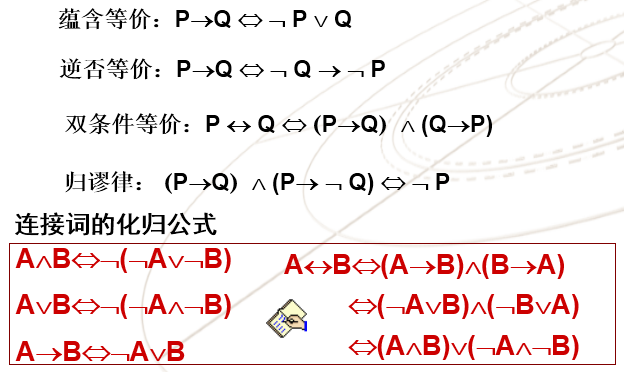
（3）附加前提法，针对结论形式为P->Q的情况。或者为P∨Q形式，因为P∨Q**⇔**⎤P->Q，即可通过引入⎤P作为附加前提开始证明。（本题也可使用该方法）



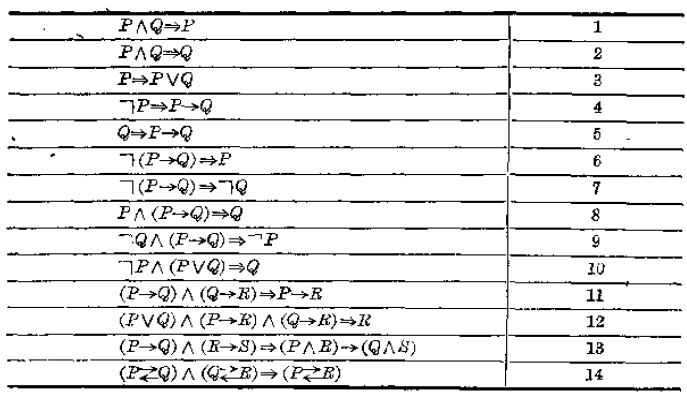
（4）任何一个证明问题，都可以采用3种方法中的任何一种，看哪种容易选用哪种即可。（如果发现直接证明不好做，就可以考虑归谬或者附加前提法）

证明过程中用到的基本等价式如下：



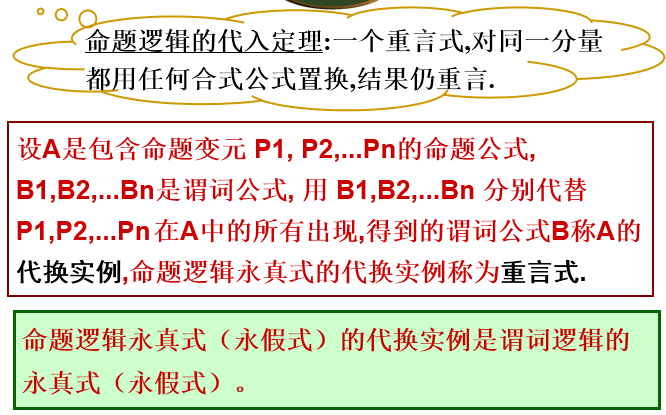


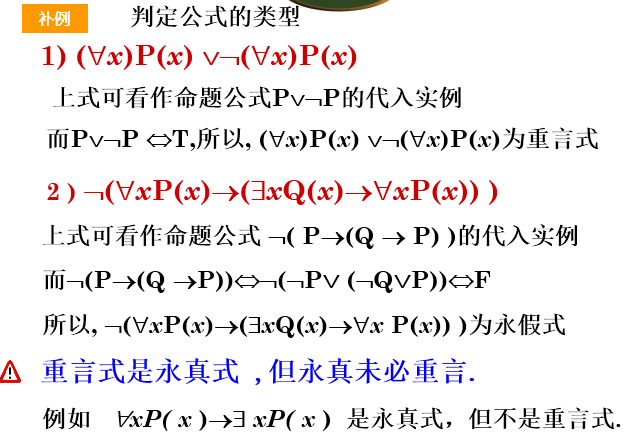
证明过程中用到的基本蕴含式如下：



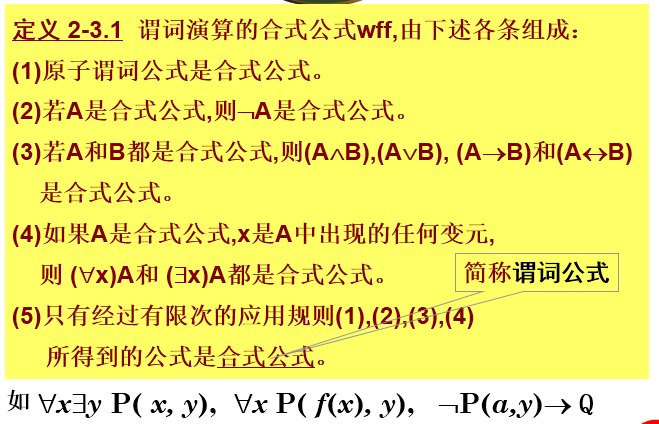
**第二部分 谓词逻辑**

**1. 谓词逻辑中永真式和重言式的差别。**

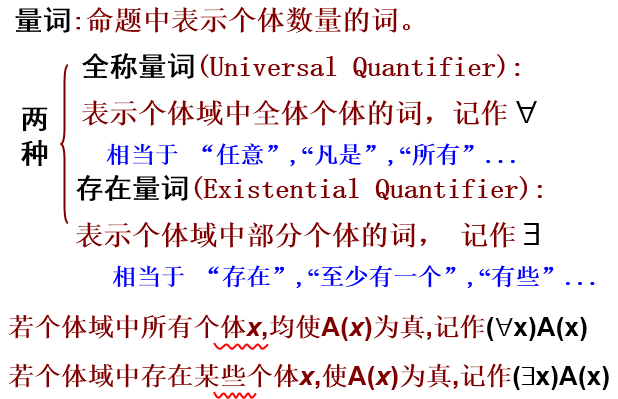




**2. 谓词公式的定义，会判定一个公式是否为谓词合式公式。**

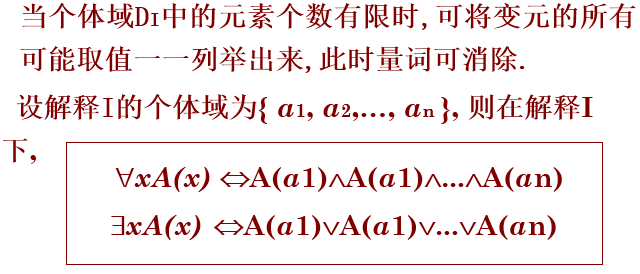


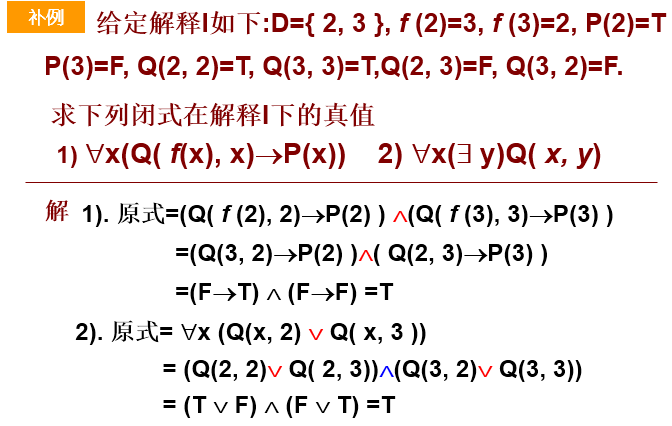
**3. 会区分量词、辖域、指导变元、约束变元、自由变元**



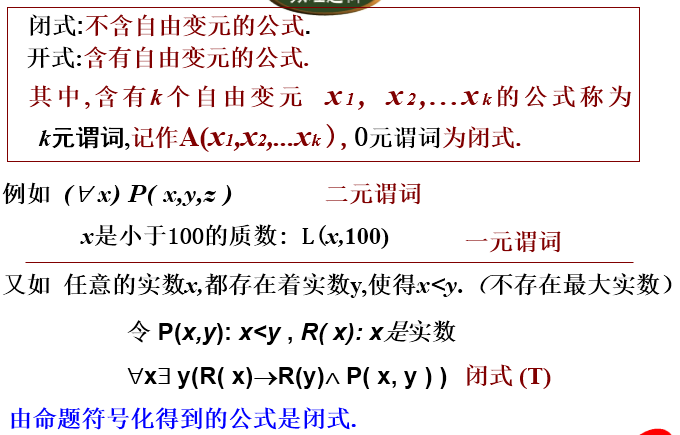


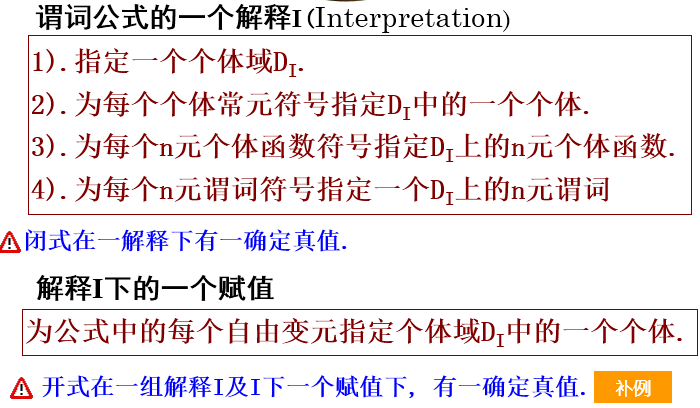
**4. 已知论域（量词的定义域），消去谓词公式中的量词（即量词的展开）。**

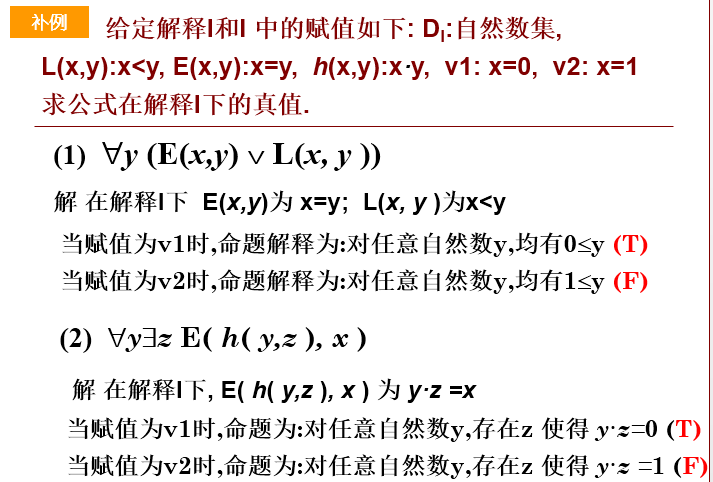


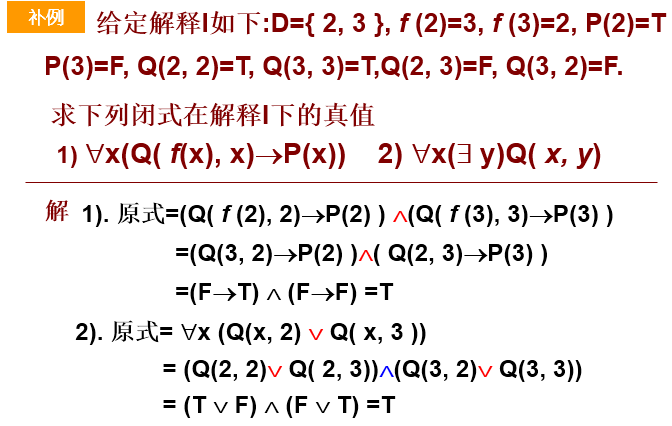


**5. 会求谓词公式的真值。**







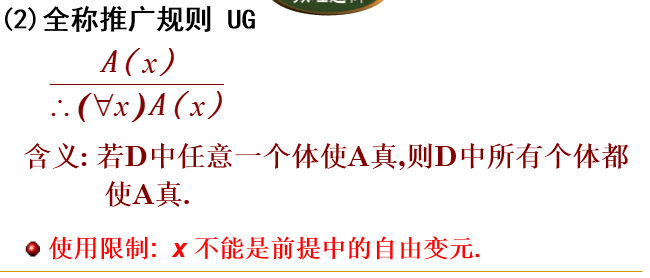


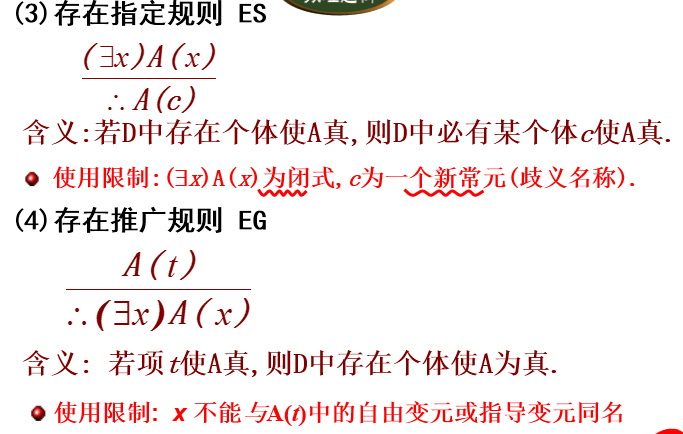
**6. 谓词的推理证明**

**考点：**谓词演算的推理理论

除了P规则（前提引入规则），T规则（结论引入规则），CP规则（附加前提规则，用于归谬法和附加前提法）外，谓词逻辑还有以下4个规则：



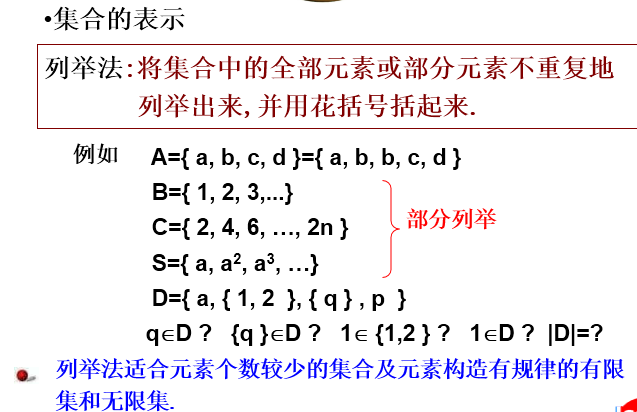


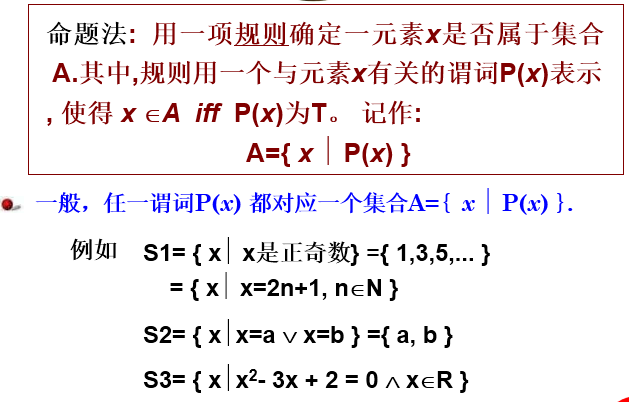


**需要注意的是：**对带有量词的谓词公式进行推理时，一定要将量词先消去才能使用相应的等价式（E）和蕴含式（I）。**原因是：**谓词演算的瑰丽方法可以看做命题演算推理方法的扩张。因为谓词演算的很多等价式和蕴含式都是命题演算相关公式的推广，所以命题演算中的推理规则P，T，CP都可以在谓词的推理理论中使用。但是在谓词推理中，某些前提与结论可能是受量词限制的，为了使用命题逻辑中的等价式和蕴含式，必须在推理过程中消去量词，进行演算，再添加量词，以使谓词演算公式的推理过程类似于命题演算重点推理。

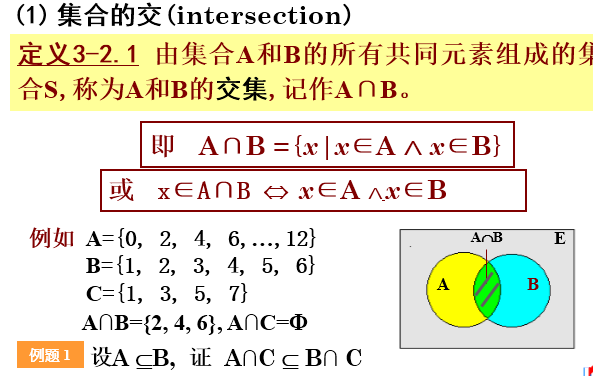
**第三部分 集合**

**1. 掌握根据集合的命题表示法，用列举法表示集合。（即根据集合的命题表示法，得到集合中的元素，再用列举法表示集合）**

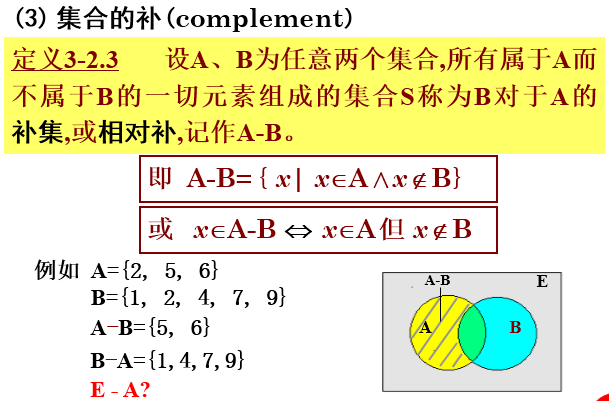




**2. 集合的运算，交、并、补（绝对补，相对补）、对称差**

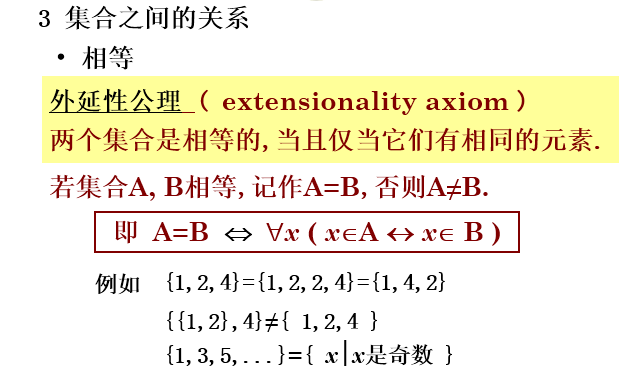


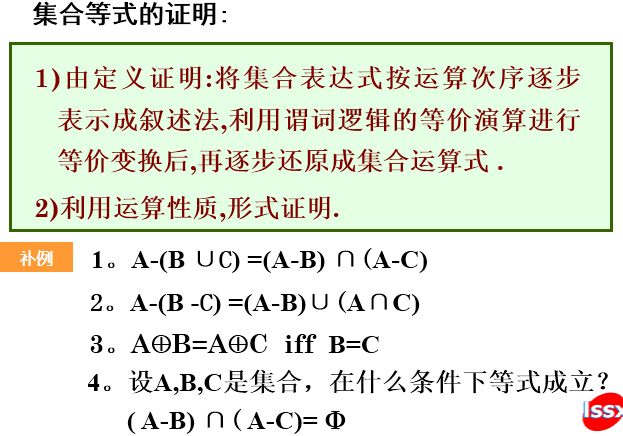




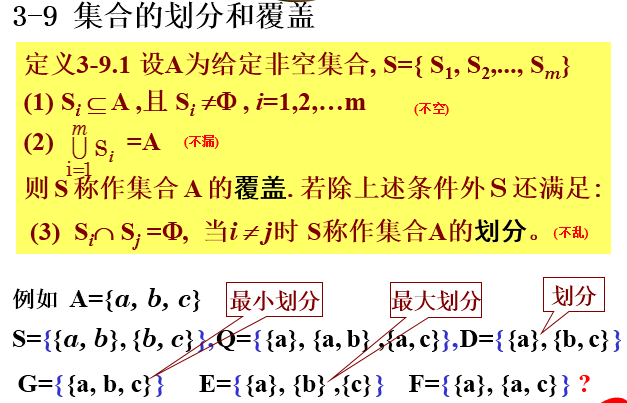


**3. 集合等式的判定，用于判定2个集合运算公式得到的结果是否相同**



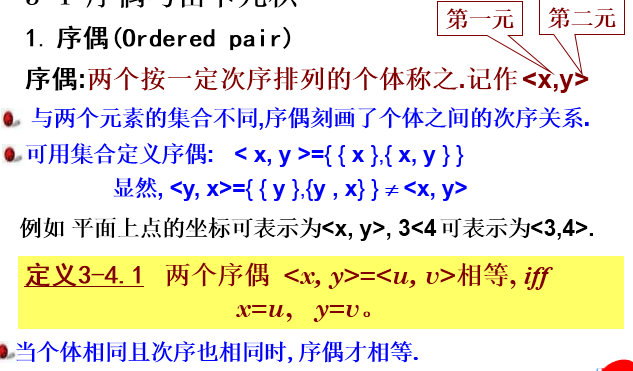


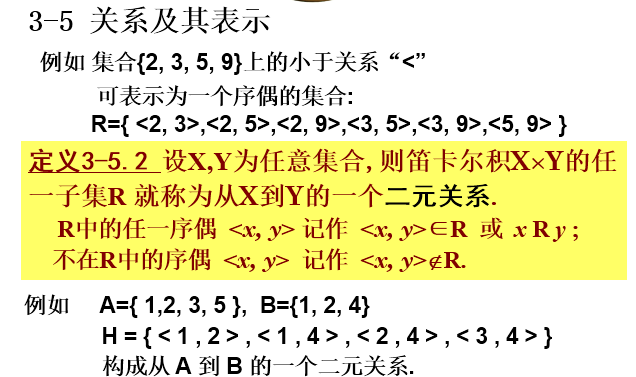
**4. 划分的定义，会求或判定集合的划分。（不空、不漏、不乱）**



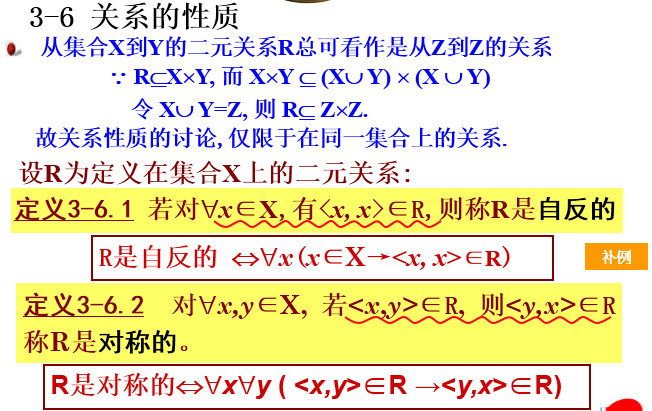


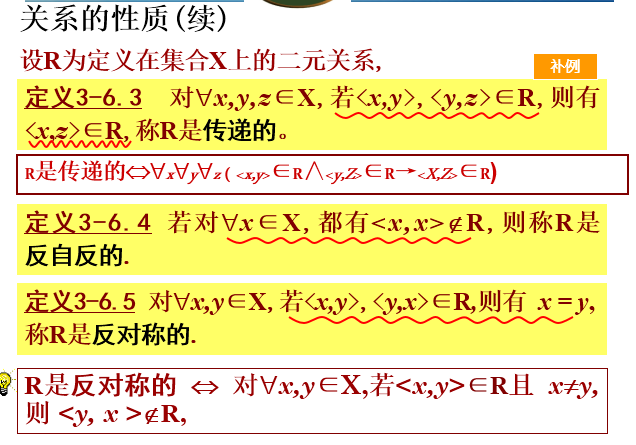
**第四部分 关系**





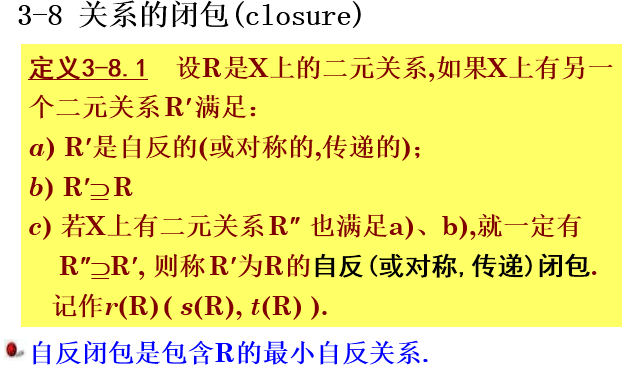
**1. 关系的性质：自反性、对称性、传递性，掌握关系性质的判定**





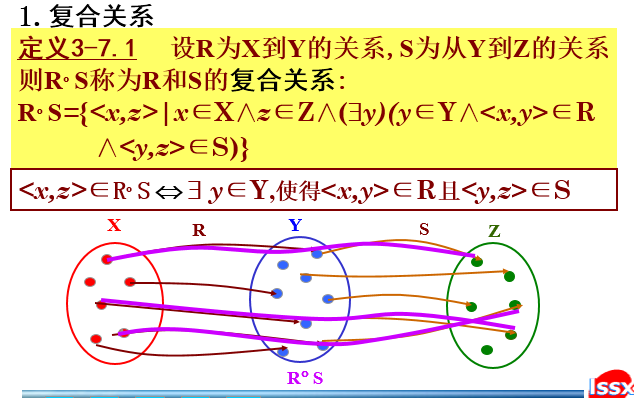
**2. 掌握关系的运算（关系是特殊的集合，由序偶组成的集合，其运算同集合的运算），并会判定运算后得到的关系的性质。**

**3. 闭包的定义：自反闭包、对称闭包、传递闭包，掌握定义及各种闭包的性质。**

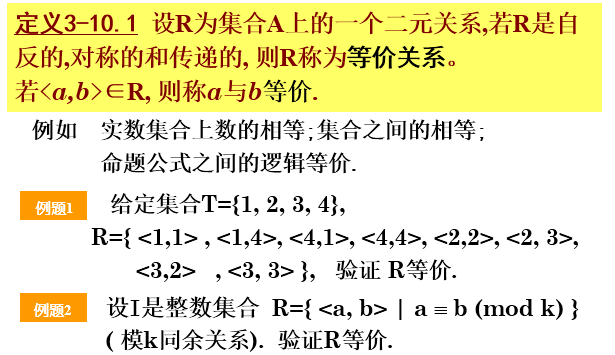


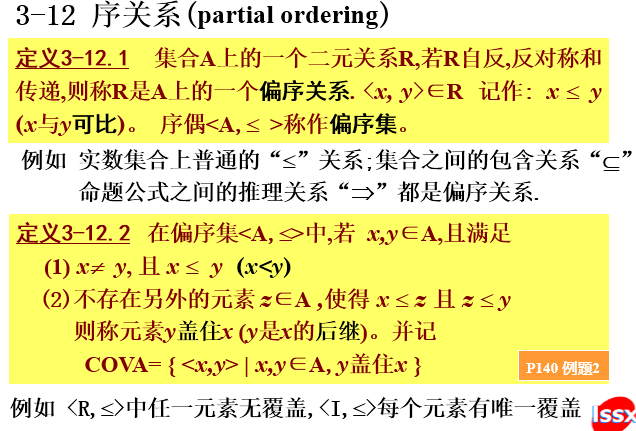
**4. 复合关系的定义，求复合关系，并会判定复合关系的性质（复合关系得到的结果仍是关系，判定方法同关系性质的判定方法）。**

**注意：求复合关系是的运算次序。**



**5. 等价关系、偏序关系的定义和判定。**

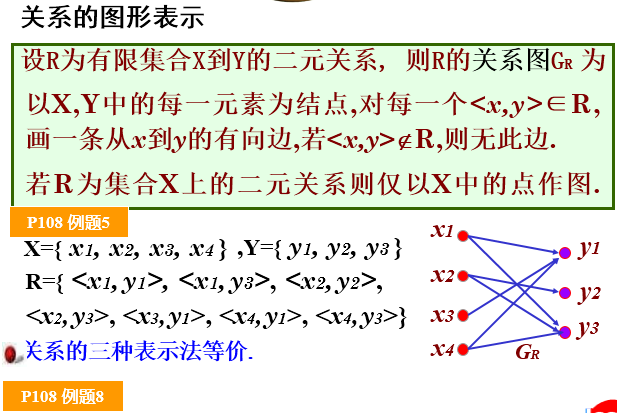




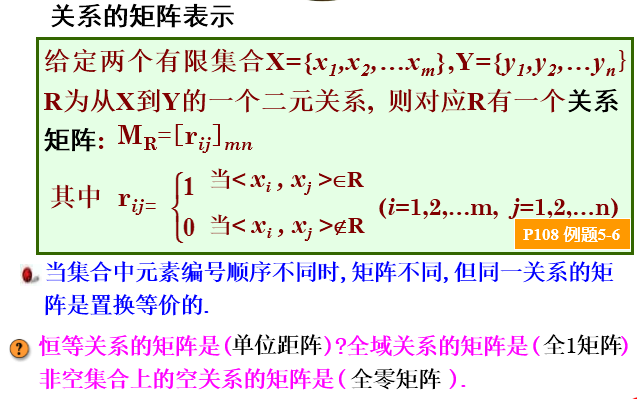
**6. 会求关系的前域（dom）和值域（ran）。**



**7. 掌握关系图的画法**



**8. 掌握关系矩阵的列举方法**



8. 根据命题描述法，用列举法表示关系。（同集合的命题法到列举法的转换）

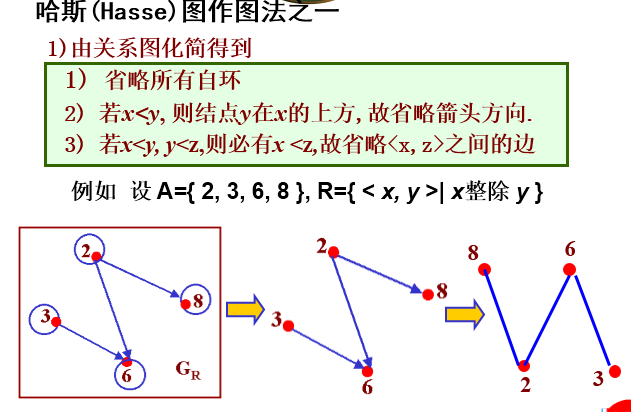
如：**给定集合 X= { 0, 1, 2, 3},**

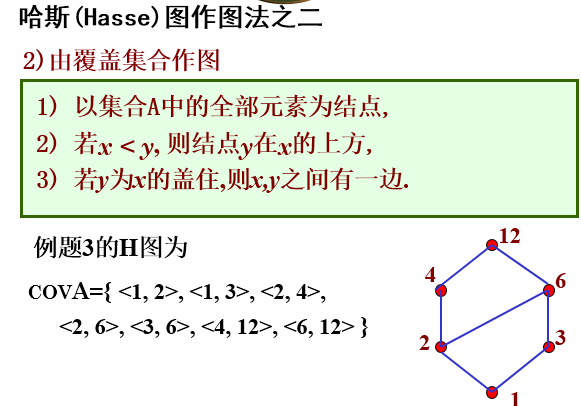
**R1={ <*i, j***>| j=i+1或 j=1/2 *i* },

R2={ <*i, j* >| i= j+2 },

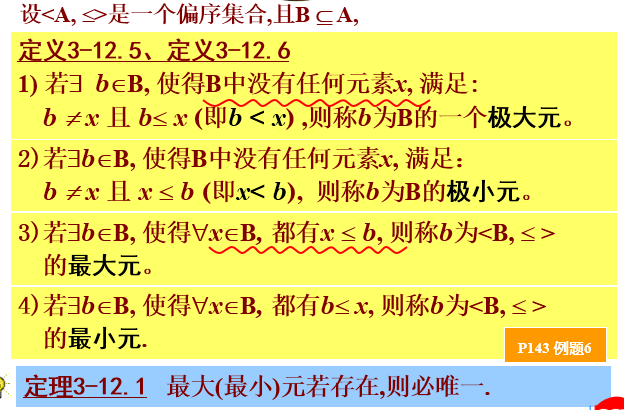
求R1和R2的列举法表示。

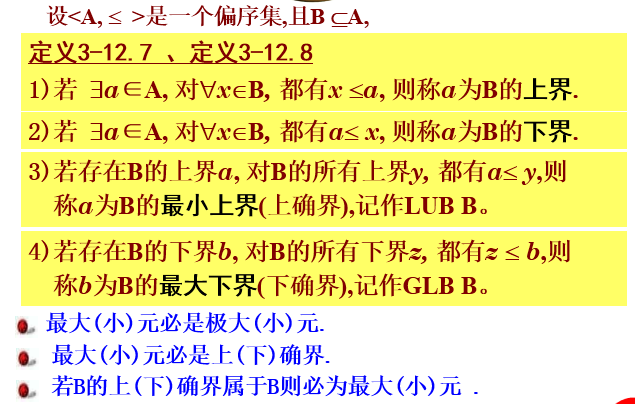
**9. 掌握哈斯图的绘制方法**





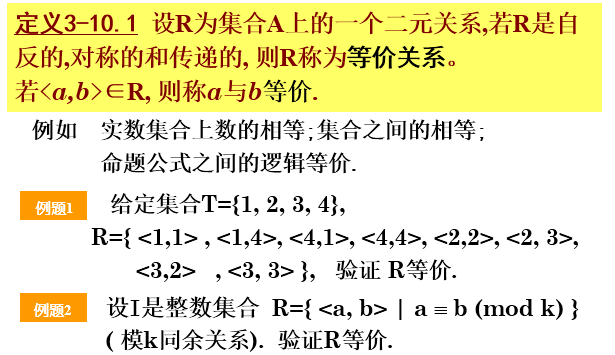
**10. 会求极大元、极小元、最大元、最小元、上界、下界、上确界、下确界。**

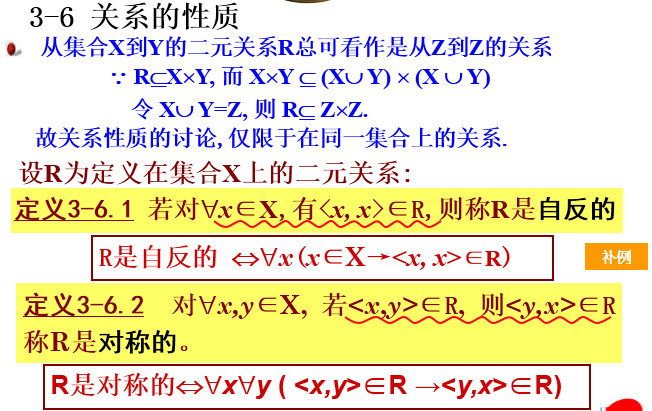


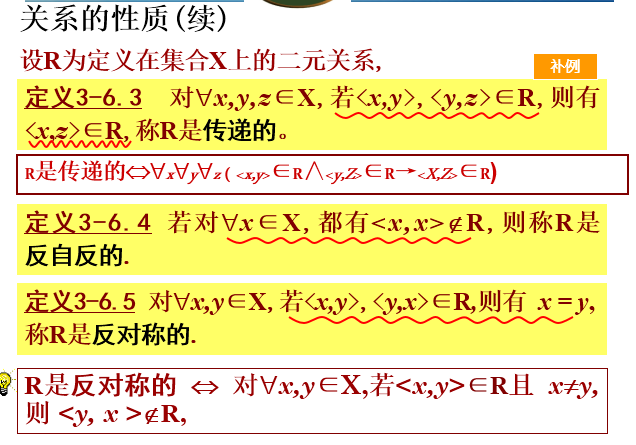


**11. 关系性质的证明，如证明关系R具有自反性；或证明关系R是等价关系等。**

**（根据等价关系的定义证明，或分别根据自反性、对称性、传递性的定义证明）**

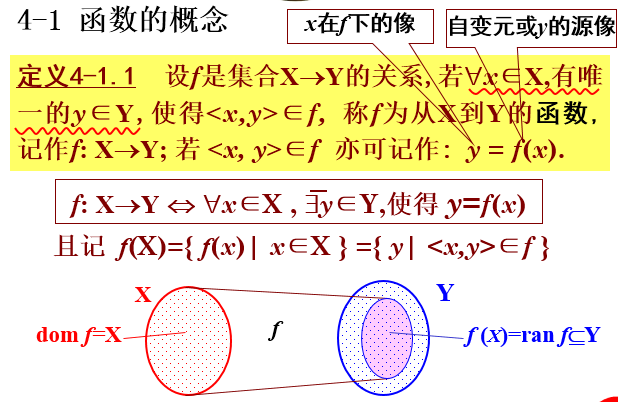


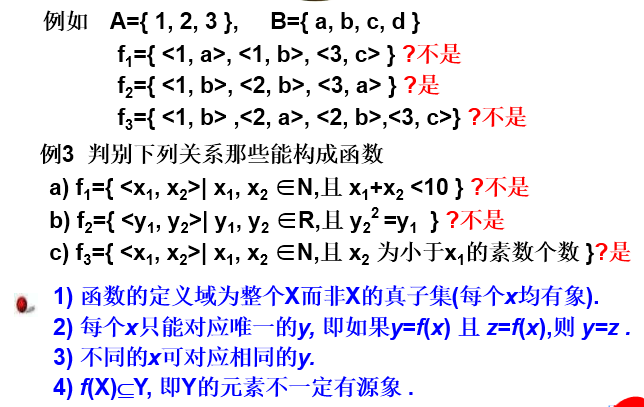




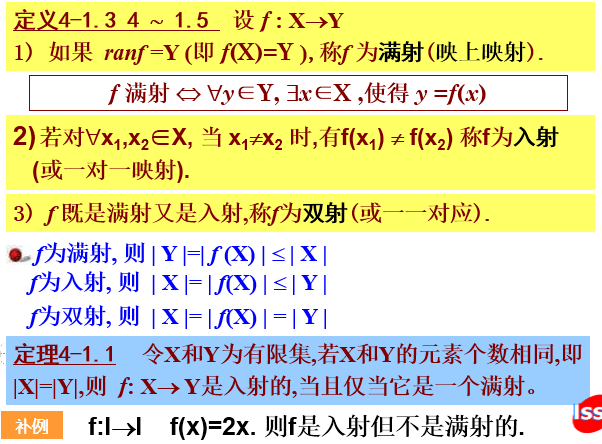
**第五部分 函数**

**1. 函数的定义，会根据定义判定一个关系是否为函数。**





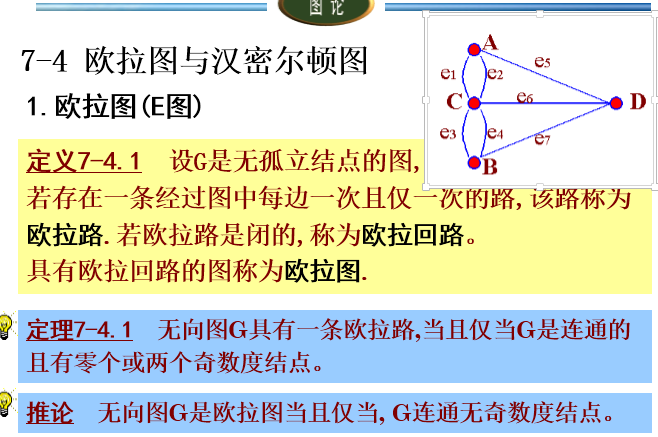
**2. 会判定函数的类型，入射函数、满射函数、双射函数**

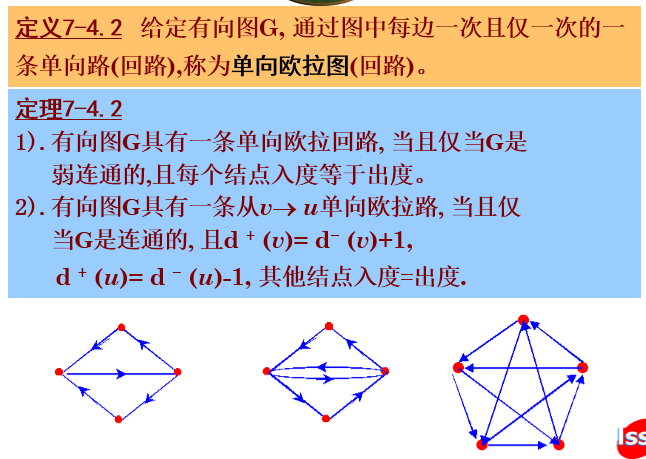


课后习题

**第六部分 图论**

**1. 掌握欧拉图的定义，理解判定欧拉图的充要条件，会判定欧拉图。**



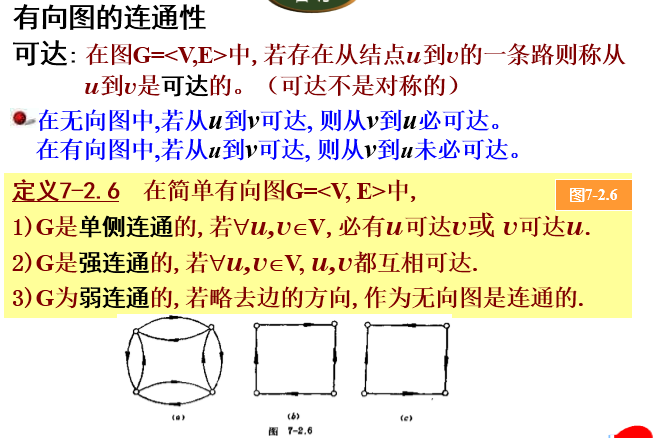


**2. 掌握汉密顿图的定义，会判定汉密顿图。**



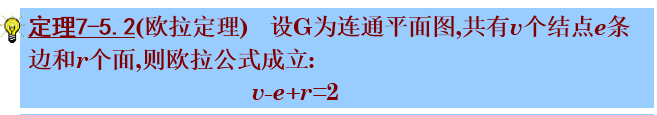
**3. 对于有向图，掌握强连通、单侧连通、弱连通的定义，并会判定强连通图、单侧连通图和弱连通图。**

**（具有强连通性的有向图是强连通图，具有单侧连通性的有向图是单侧连通图，具有弱连通性的有向图是弱连通图）**



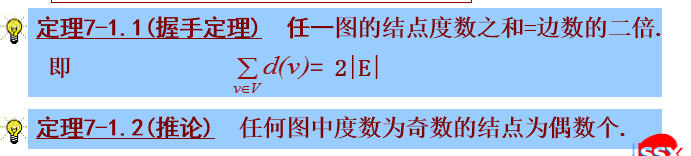
**4. 会根据给定的图的集合表示法G=<V,E>,V={….},E={….}，绘制有向图和无向图。**

**5. 掌握平面图的欧拉定理**

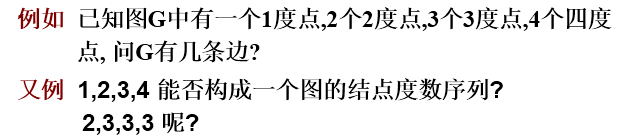


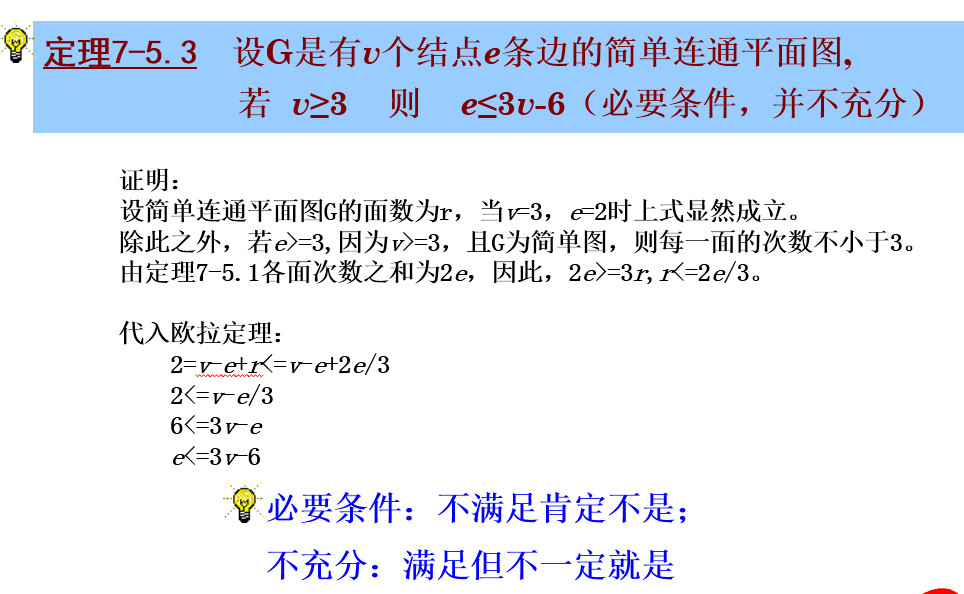
****

**6. 掌握图中结点的度和边之间的量的关系。**

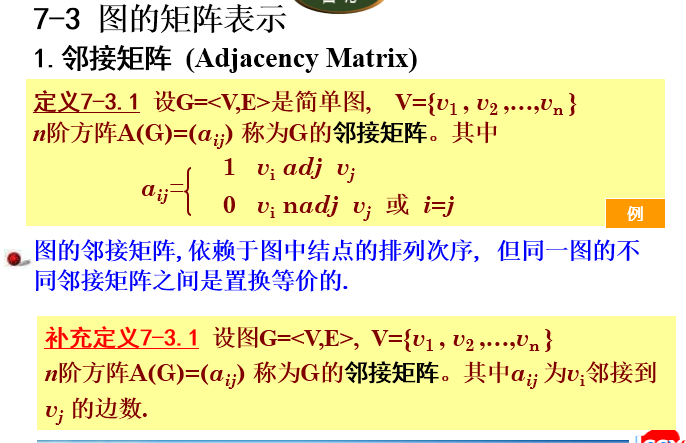


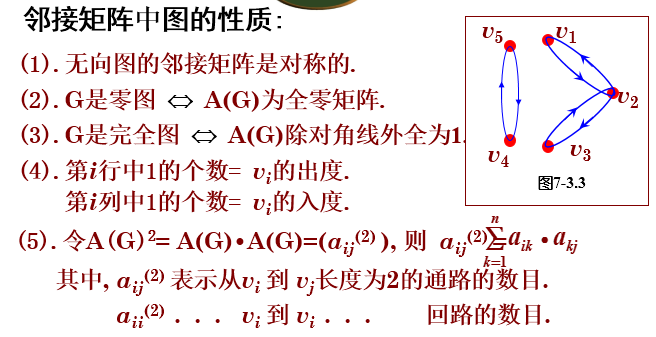






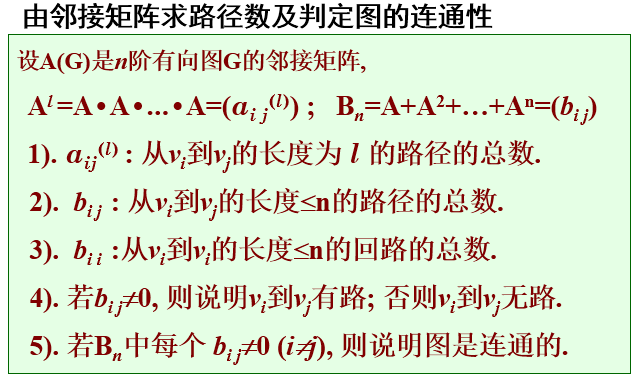
**7. 掌握邻接矩阵的绘制方法，并掌握邻接矩阵中行、列等数值代表的含义。**



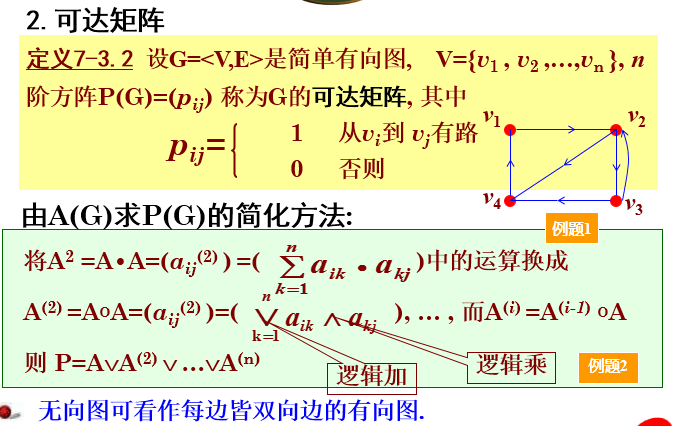


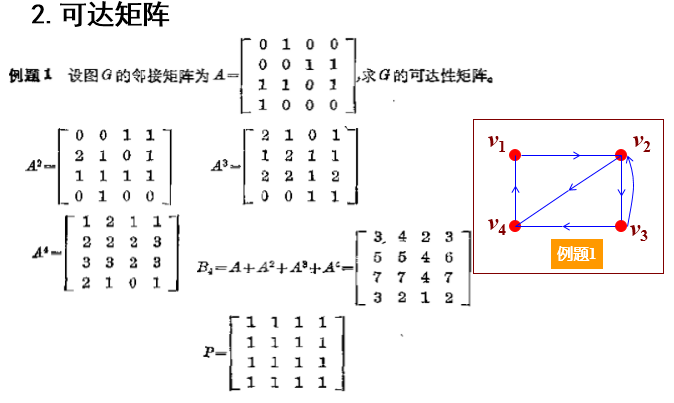
**8. 会求路径路径长度为n（n<=|V|）的路的条数，利用邻接矩阵求解，求An。如n=2，即求v1到v2的路的长度为2的路有多少条？即求A2，其中a12的值即为长度为2的路径的数量。**



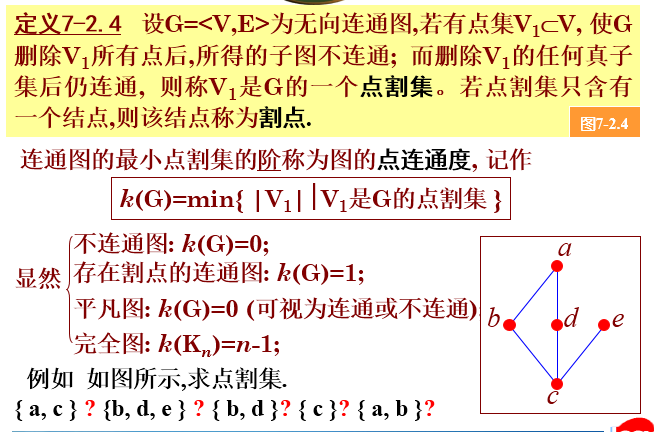


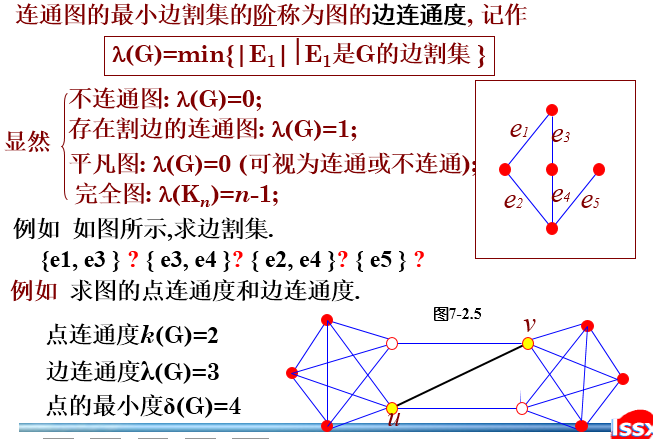
**9. 掌握可达矩阵的计算方法。**



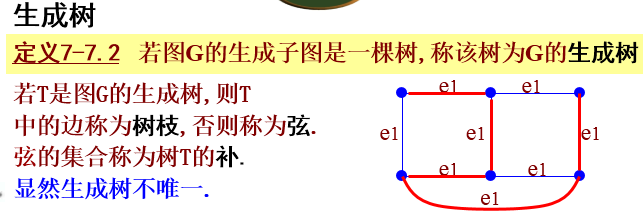


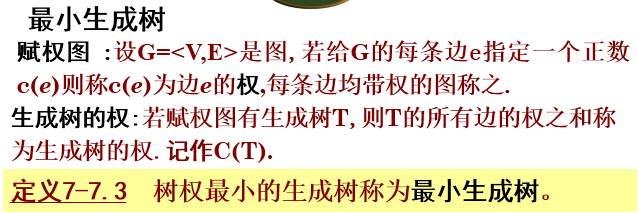
**10. 掌握点割集，割点、边割集、割边的概念，并会求图的点割集，割点、边割集、割边。**

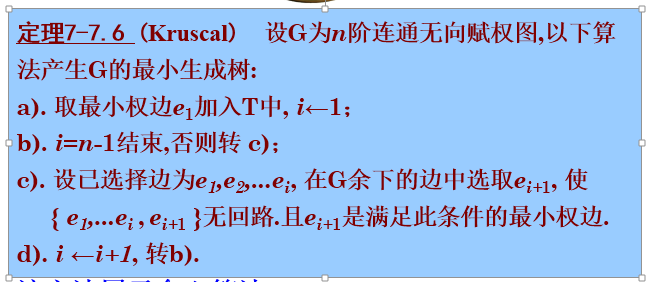




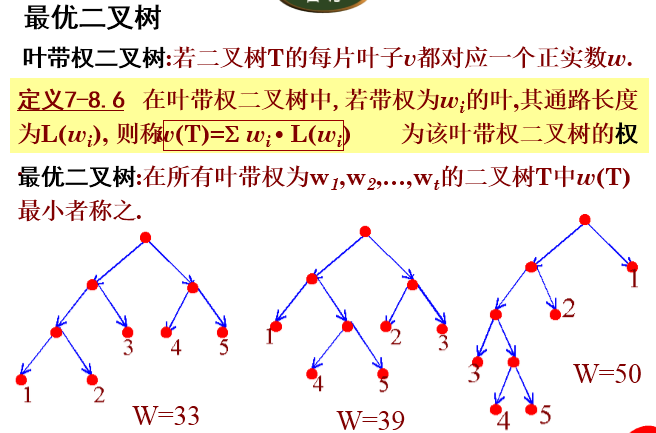
**11. 会求图的最小生成树，即最小生成树的权值**

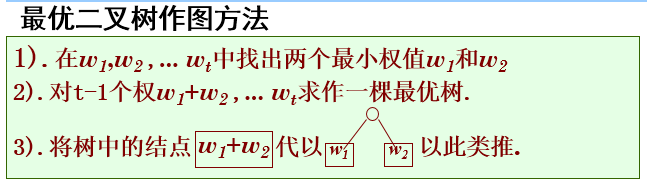


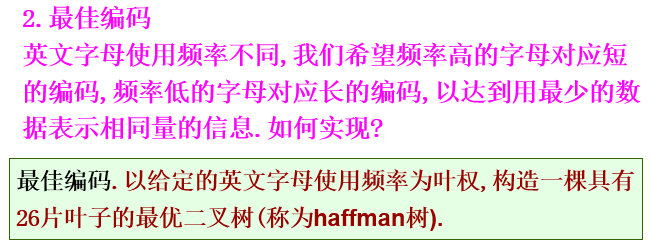


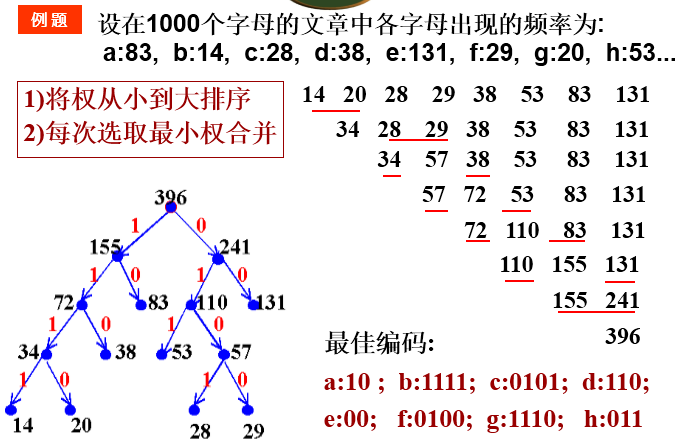


**12. 会求最优二叉树，并能根据最优二叉树给出最优编码**









**13会画对偶图，并给对偶图的顶点着色**

