四. system specifications 系统规范说明

S₁. 可满足的(consistent)

当系统规范说明中不存在矛盾,即不会导出矛盾时,我们称此系统规范说明是可满足的。

当我们无法开发出一个满足所有规范说明的系统时,我们称此系统规范说明不可满足.

1.2 propositional equivalences 命题等价

一、introduction 介绍

D₁. 当一个复合命题无论其中任一原子命题的真值是真是假,总是真的,则此复合命题为重言式(永真式);当一个复合命题无论其中任一原子命题的真值是真是假,其总是假的,则此复合命题为矛盾式(永假式);当一个复合命题无论其中任一原子命题的真值是真是假,其有真有假,则此复合命题为不定式.(既不是重言式(永真式)的命题,又不是矛盾式(永假式)的命题为不定式命题)

二. logical equivalences 逻辑等价

S₁. 逻辑等价(logical equivalences):

当两个或多个复合命题不论在什么情况下总有相同真值时,我们称这些命题是逻辑等价的.

D₁. 当命题 $p\leftrightarrow q$ 为永真式时,我们称复合命题 p、q 是逻辑等价的,记为 $p\equiv q$

S₂. 德摩根律(De Morgan laws)

由19世纪英国数学家德摩根所命名的两条逻辑等价定律. $\neg(p\vee q)\equiv\neg p\wedge\neg q$; $\neg(p\wedge q)\equiv\neg p\vee\neg q$

D₂. 逻辑等价表:

恒等律:	$p\wedge T\equiv p$	$p\vee F\equiv p$
零律:	$p\vee T\equiv T$	$p\wedge F\equiv F$
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:	$(p\vee q)\vee r\equiv p\vee(q\vee r)$	$(p\wedge q)\wedge r=p\wedge(q\wedge r)$
分配律:	$p\vee(q\wedge r)\equiv(p\vee q)\wedge(p\vee r)$	$p\wedge(q\vee r)\equiv(p\wedge q)\vee(p\wedge r)$
德摩根律:	$\neg(p\wedge q)\equiv\neg p\vee\neg q$	$\neg(p\vee q)\equiv\neg p\vee\neg q$
吸收律:	$p\vee(p\wedge q)\equiv p$	$p\wedge(p\vee q)\equiv p$
否定律:	$p\vee\neg p\equiv T$	$p\wedge\neg p\equiv F$

D3. 包含条件语句的重要逻辑等价

$$p \to q \equiv \neg p \vee q$$

$$p \leftrightarrow q \equiv (p \to q) \wedge (q \to p)$$

三. constructing new logical equivalences 构造新的逻辑等价

M1. 证明命题逻辑等价的方法

① 可以通过列真值表，证明在任何情况下真值相同(适合不太复杂的命题)

② 通过已有的逻辑等价式进行推导(适合比较复杂的命题)

1.3 predicates and quantifiers 谓词与量词

一. introduction 介绍

S1. ~~pre~~ 谓词逻辑(predicate logic)

一种更为强大的逻辑类型，它可以通过允许我们论证与探索对象之间的联系，从而可以表达在数学及计算机科学上出现的各种各样许许多多的命题.

二. predicates 谓词

S1. 谓词(predicates)

说明语句主语(即对象)所具有的性质

S2. 命题函数(propositional function)

用以描述谓词逻辑的一种函数，其表示了命题的一般形式. 命题函数在严格意义上并不能称为命题，只有当其中的变量被确定时，该语句才能成为命题，并有确定的真值.

S3. n元谓词(n-place predicate, n-ary predicate)

当命题函数中有多个变量时，此时命题函数也称n元谓词

S4. 先决条件(preconditions)

在计算机科学中计算机对于给定的有效~~輸~~输入总能产生一个所需输出. 其中描述有效输入的语句称为先决条件

S5. 后置条件(postconditions)

在程序运行过程中输出所要满足的情况称为后置条件

三、quantifters 量词

S₁. 量化（quantification）

从命题函数中创建一个新~~函数~~命题的过程称量化

S₂. 谓词演算（predicate calculus）

处理谓词与量词的逻辑领域

S₃. 论域（domain of discourse / universe of discourse），领域（domain）

许多的数学命题具有对于在一个特定的数域内的所有的数该命题总是真的的性质。这一特定的数域，我们称其为论域，也称作领域。

D₁. 对于命题函数 $P(x)$ 而言，其全称~~命题~~量化为：~~对于域中任意的值~~ $P(x)$，其中 x 为域中任意的值。我们将 $P(x)$ 的全称量化记为 $\forall x P(x)$。在这里 $\forall$ 称为全称量词。我们将 $\forall x P(x)$ 读作对于所有的 x，$P(x)$ 或读作对于每一个的 x，$P(x)$。使得 $P(x)$ 为假的元素称为 $\forall x P(x)$ 的反例。

S₄. 全称量词（universal quantifter）

表示"所有的"、"每一个的"的含意，记作 $\forall$

S₅. 反例（counterexample）

使得命题函数 $P(x)$ 为假的元素称为 $\forall x P(x)$ 的反例

R₁. 一般情况下我们默认全称量词所对应的论域不为空。因为一旦论域为空，$\forall x P(x)$ 对于任意的命题函数 $P(x)$ 总成立，因为在论域中没有一个元素 x 证明 $P(x)$ 为假

R₂. 最好不要出现"对于任意的 x"，因为这样表达是模棱两可的，我们搞不清这里"任意的"指的是"每一的"还是"某些的"。当然，有些时候却是准确的。

R₃. 对于一个命题函数 $P(x)$ 而言，语句 $\forall x P(x)$ 为假当且仅当 $P(x)$ 对于域内的 x 值不总是真的。对于证明 $P(x)$ 对于域内 x 值不总是真的的一种方法是找出 $\forall x P(x)$ 的反例

D₂. 对于命题函数 $P(x)$ 而言，其特称量化为：$P(x)$，其中 x 为域中存在的某一元素。我们将 $P(x)$ 的特称量化记为 $\exists x P(x)$。在这里 $\exists$ 称为特称量词。

S₆. 特称量词（existential quantifter）

表示"某一的"的含意，记作 $\exists$

D₃.

语句	真	假
$\forall x P(x)$	$P(x)$ 对于每一个 x 均为真	存在一个 x 使 $P(x)$ 为假
$\exists x P(x)$	存在一个 x 使 $P(x)$ 为真	$P(x)$ 对于每一个 x 均为假

R₄. 无论对于全称量词还是特称量词而言，在使用时都应标明域的范围。当域变化时，$\forall x P(x)$（$\exists x P(x)$）~~的~~真值可能发生改变。不标明域的范围是无意义的。

R_5. 像全称量化一样，特称量化的域默认非空。因为如果域是空的，对于任意的命题函数 $Q(x)$，$\exists x Q(x)$ 总是假的，因为没有在域中的任一元素可证明 $Q(x)$ 为真。

四、other quantifiers 其他量词

S_1. 唯一性量词（uniqueness quantifier）

用于表达存在且唯一的量词（即：有且仅有一个），记为 $\exists!$ 或 $\exists_1$

R_1. 由于唯一性量词可通过之前所学的谓词与量词进行表达，因而较少被使用

五、quantifiers with restricted domains 限定域下的量词

R_1. 对于全称量化在特定域下的限制相当于全称量化加上一个条件语句，即：

$\forall x p(x)\,(Q(x)) \equiv \forall x\,(p(x) \to Q(x))$　例：$\forall x<0\,(x^2>0) \equiv \forall x\,(x<0 \to x^2>0)$

对于存在量化在特定域下的限制相当于存在量化~~加上~~的析取，即：

$\exists x\, p(x)\,(Q(x)) \equiv \exists x\,(p(x) \vee Q(x))$　例：$\exists x<0\,(x^2>0) \equiv \exists x\,(x<0 \vee x^2>0)$

六、precedence of quantifiers 量词的优先级

R_1. 量词的优先级高于其他任何一个逻辑运算符

七、logical equivalences involving quantifiers 包含量词的逻辑等价

D_1. 包含谓词、量词的语句逻辑等价当且仅当无论任何谓词替代进语句中与任何论域用于该命题函数作为变量。我们使用记号 $S \equiv T$ 表示包含谓词量词的语句 S、T 是逻辑等价的。

R_1. 常见的两种逻辑等价
$\forall x\,(P(x) \wedge Q(x)) \equiv \forall x P(x) \wedge \forall x\, Q(x)$　全称量词对合取可分配
$\exists x\,(P(x) \vee Q(x)) \equiv \exists x P(x) \vee \exists x\, Q(x)$　全称量词对析取可分配

R_2. 注意以下两种形式在逻辑上不等价
$\forall x\,(P(x) \vee Q(x))$ 与 $\forall x P(x) \vee \cancel{Q(x)}\ \forall x Q(x)$
$\exists x\,(P(x) \wedge Q(x))$ 与 $\exists x P(x) \wedge \exists x Q(x)$

八、negation quantified expressions 否定量化表达

S_1. 对于量词的德摩根律：

否定	等价语句	当否定式为真	当否定式为假
$\neg\exists x P(x)$	$\forall x \neg P(x)$	对于每一个 x，P(x) 为假	存在一个 x 使 P(x) 为真
$\neg\forall x P(x)$	$\exists x \neg P(x)$	存在一个 x 使 P(x) 为假	对于每一个 x P(x) 均为真

R_1: $\neg(P(x_1) \wedge P(x_2) \wedge \cdots \wedge P(x_n)) \equiv \neg P(x_1) \vee \neg P(x_2) \vee \cdots \vee \neg P(x_n)$　相当于 $\neg\forall x P(x) \equiv \exists x \neg P(x)$

$\neg(P(x_1) \vee P(x_2) \vee \cdots \vee P(x_n)) \equiv \neg P(x_1) \wedge \neg P(x_2) \wedge \cdots \wedge \neg P(x_n)$　相当于 $\neg\exists x P(x) \equiv \forall x \neg P(x)$

No.

Date

1.4 nested quantifiers 量词的嵌套

一、introduction 介绍

S₁. 量词嵌套(nested quantifiers)

一个量词如果在另一个量词的作用域内，称这个量词是被嵌套的

二、the order of quantifiers 量词的顺序

R₁. 两个变量的量化

语句	何时为真	何时为假
$\forall x \forall y P(x,y)$ $\forall y \forall x P(x,y)$	$P(x,y)$对于x,y的任意组合均为真	存在一个x,y的组合使$P(x,y)$为假
$\forall x \exists y P(x,y)$	对于任意的x存在一个y使$P(x,y)$为真	存在任一的x，对于任意的y，$P(x,y)$均为假
$\exists x \forall y P(x,y)$	存在任一的x，对于任意的y，$P(x,y)$均为真	对于任意的x存在一个y使$P(x,y)$为假
$\exists x \exists y P(x,y)$ $\exists y \exists x P(x,y)$	存在一个x,y的组合使$P(x,y)$为真	$P(x,y)$对于x,y的任意组合均为假

三、negating nested quantifiers 嵌套量词的否定

R₁. 一般还是采用德摩根律的方法由外向内进行否定

例：$\neg\exists w \forall a \exists f (P(w,f) \land Q(f,a))$ ………… 把$\forall a \exists f(P(w,f)\land Q(f,a))$看作一个命题函数

$\equiv \forall w \neg \forall a \exists f(P(w,f)\land Q(f,a))$ ………… 把$\exists f(P(w,f)\land Q(f,a))$看作一个命题函数

$\equiv \forall w \exists a \neg \exists f(P(w,f)\land Q(f,a))$ ………… 把$(P(w,f)\land Q(f,a))$看作一个命题函数

$\equiv \forall w \exists a \forall f \neg(P(w,f)\land Q(f,a))$ ………… 德摩根律

$\equiv \forall w \exists a \forall f (\neg P(w,f) \lor \neg Q(f,a))$

1.5 rules of inference 推理规则

一、introduction 介绍

S₁. 论证(argument)

以结论结尾的一连串语句

S₂. 有效的(valid)

论证最终得出的结论必须由之前的所有语句的真实事实中推断出来，即，如果一个论证是有效的当且仅当该过程中不出现前提为真但结论为假的情况

S3. 前提(premises)

在论证中用于推出结论的语句

S4. 谬论(fallacies)

错误的，导致无效论证的论证形式

二、valid arguments in propositional logic 命题逻辑中的有效论证

R1. 当一个论证中，前提全为真且结论也为真的论证形式是有效的

S1. 论证形式(argument form)

我们为分析一个论证把命题替换成对应的命题变元，这也将论证转换成了论证形式。论证的有效性取决于论证形式的有效性

D1. 在命题逻辑里，论证是一连串的命题。除了最后一个命题外的所有命题称为前提，最后一个命题称为结论。当论证中前提全为真，结论也为真时，我们称此论证是有效的。

在命题逻辑里，论证形式是包含命题变元的一连串组合命题。当无论哪个特定的命题替换此论证形式中的命题变元，且当它的前提为真时，结论也为真时，我们称此论证形式是有效的。

R2. 在命题逻辑中证明论证有效性的关键是要证明其论证形式的有效性，而不是证明结论的正确性。有时即使结论为假，只要论证形式为有效的(即此时的前提为假)，这一论证仍是有效的。

三、rules of inference for propositional logic 命题逻辑中的推理规则

S1. 推理规则(rules of inference)

一些已被证明的简单基础的论证形式。这些推理规则可被用于构造更为复杂的有效的论证形式

S2. 分离规则(modus ponens / law of detachment)

由命题 p 与命题 $p\to q$ 推出命题 q 的一种推理规则

R1. 推理规则

推理规则	重言式/永真式	命名
p $p\to q$ $\therefore q$	$[p\wedge(p\to q)]\to q$	分离规则
$\neg q$ $p\to q$ $\therefore \neg p$	$[\neg q\wedge(p\to q)]\to\neg p$	拒取式
$p\to q$ $q\to r$ $\therefore p\to r$	$[(p\to q)\wedge(q\to r)]\to(p\to r)$	假言三段论
$p\vee q$ $\neg p$ $\therefore q$	$[(p\vee q)\wedge\neg p]\to q$	析取三段论

(后续)

(续前) 推理规则	重言式/永真式	命名
p $\therefore p \vee q$	$p \to (p \vee q)$	附加律
$p \wedge q$ $\therefore p$	$(p \wedge q) \to p$	化简律
p q $\therefore p \wedge q$	$[(p) \wedge (q)] \to (p \wedge q)$	合取律
$p \vee q$ $\neg p \vee r$ $\therefore q \vee r$	$[(p \vee q) \wedge (\neg p \vee r)] \to (q \vee r)$	消去律/消解律/归结规则

四、resolution 归结

S1. 归结(resolution)

一种基于永真式$[(p \vee q) \wedge (\neg p \vee r)] \to (q \vee r)$的推理规则. 此推理规则被程序广泛采用.

S2. 消去式/归结式(resolvent)

归结规则中最后所得结论的那个包含析取的命题称为消去式(归结式)

S3. 子句(clauses)

变元的析取或变元的否定形式. 在归结规则中, 假设与结论必须被表示为子句的形式. 非子句的语句可用一个或多个子句的语句替换

五、fallacies 谬论

S1. 肯定结论谬论(fallacy of affirming the conclusion)

从$p \to q$中认为q为真亦能推出p为真的错误论证形式

S2. 否定前提谬论(fallacy of denying the hypothesis)

从$p \to q$中认为$\neg p$为真亦能推出$\neg q$为真的错误论证形式

R1. 谬论是由于采用了错误的论证形式与推理规则. 结论错误并不能说明论证是无效的, 因而不能说明谬论的产生.

六、rules of inference for quantified statements 含有量词的语句的推理规则

S1. 全称量词消去/全称量词例化/UI规则(universal instantiation)

在$\forall x P(x)$为真的前提下, 若c在x的论域中, 可得结论$P(c)$为真的推理规则

S2. 全称量词引入/全称量词生成/UG规则(universal generalization)

在x的论域中的任一元素c都可使$P(c)$为真的前提下, 可得结论$\forall x P(x)$为真的推理规则

S3. 存在量词消去 / EI规则(existential instantiation)

在$\exists x P(x)$为真的前提下，可得结论在x的论域中存在某一元素c使P(c)为真的推理规则

S4. 存在量词引入 / EG规则(existential generalization)

在x的论域中存在一元素c使P(c)为真的前提下，可得结论$\exists x P(x)$为真的推理规则

R1.

推理规则	命名
$\dfrac{\forall x P(x)}{\therefore P(c)}$	全称量词消去/全称量词例化/UI规则
$\dfrac{P(c)\ (c为论域中任意值)}{\therefore \forall x P(x)}$	全称量词引入/全称量词生成/UG规则
$\dfrac{\exists x P(x)}{\therefore P(c)\ (c为论域x中某一值)}$	存在量词消去/EI规则
$\dfrac{P(c)\ (c为论域x中某一值)}{\therefore \exists x P(x)}$	存在量词引入/EG规则

k. combining rules of inference for propositions and quantified statements 命题推理和量化语句推理规则

S1. 全称分离定律(universal modus ponens)

由$\forall x(P(x)\to Q(x))$与P(a)(a为论域中某一特定元素)可推出Q(a)的一种推理规则

$\forall x(P(x)\to Q(x))$

P(a)，a为论域中某一特定元素

$\therefore Q(a)$

S2. 全称拒取式(universal modus tollens)

由$\forall x(P(x)\to Q(x))$与$\neg Q(a)$(a为论域中某一特定元素)可推出$\neg P(a)$的一种推理规则

$\forall x(P(x)\to Q(x))$

$\neg Q(a)$，a为论域中某一特定元素

$\therefore \neg P(a)$

No.

第二章：基本结构：集合、函数、数列与求和

Chapter 2: Basic Structures: Sets, Functions, Sequences, and Sums

2.1 sets 集合

一、introduction 介绍

~~S~~ R1、集合是构成其他离散结构的基础离散结构

R2、集合语言为一种研究这一系统组合的方法

D1、集合是对象的无序组合

R3、集合是最为基本的概念

S1、悖论(paradoxes)

直观朴素的集合定义中由于对对象没有加以限定，因而导致悖论的产生.其中罗素悖论较为著名。在罗素提出一个"集合"(至少在当时定义下为集合) $S=\{x|x\notin x\}$，在S中由于 $\phi\notin\phi$，$N\notin N$，$Z\notin Z$…，因而S包含所有其他的集合.但如果S是集合，则我们可以判断出哪些是集合的元素，哪些不是.那么S是不是S的元素？若是，则 $S\notin S$；若不是则 $S\in S$，从而出现矛盾.因而 $S=\{x|x\notin x\}$ 并不是集合

S2、朴素集合论(naive set theory)

由康托尔所提出，很好地避免了集合定义~~中的~~所引发的矛盾

D2、集合中的对象称为元素，或成员.集合与元素间是包含关系.

R4、~~符号~~ "$a\in A$"代表a为集合A中的一个元素；"$a\notin A$"代表a不为集合A中的一个元素

R5、集合一般用大写字母表示，元素一般用小写字母表示

R6、描述集合的其中一种方法为列出集合中所有元素。如果元素间有规律，可以列出几个后加上省略号表示剩余元素

S3、集合结构(set builder)：

另一种描述集合的方法是使用集合结构记号。我们通过描述集合中元素的特性来定义集合

S4、自然数(natural numbers)

~~由~~自然数集为：$N=\{0,1,2,3,\ldots\}$ (注：0是否属于自然数仍有一定争议)

S5、整数(integers)

整数集为：$Z=\{\ldots,-2,-1,0,1,2,\cdots\}$

S6、正整数(positive integers)

正整数集为：$Z^+=\{1,2,3,\ldots\}$

S7、有理数(rational numbers)

有理数集为：$Q=\{p/q|p\in Z, q\in Z, ~~and~~\ q\neq 0\}$

S8、实数(real numbers)

实数集为R

R7、注意下面这个集合$\{N\}$，对于N而言，$N \in \{N\}$；但对于1而言，$1 \notin \{N\}$

S9、数据类型(datatype)，类型(type)

在计算机科学中，数据类型/类型的概念基于集合的概念．特别地，数据类型/类型是集合的名称，并伴有一系列可对集合内对象进行的一系列操作．

D3．两个集合相等当且仅当两个集合包含相同的元素．即，若A，B均为集合，集合A，B相等当且仅当命题$\forall x(x \in A \leftrightarrow x \in B)$为真．我们将集合A，B相等记为$A=B$

R8．集合相等不要求集合内元素顺序相同，也不要求元素出现次数相同，只要求包含的元素相同

S9．全集(universal set)

表示集合中所有的元素

R9．集合可以通过维恩图进行几何上的表示．在维恩图中，全集用矩形表示．在矩形中，我们可以用圆形或其他几何图形表示集合，也可以用点表示集合中的元素．维恩图通常用来表示集合之间的关系

S10，空集(empty set)(null set)

不包含任何元素的特殊集合，记作ϕ(有时也记作$\{\}$)．有些集合等于空集．

S11．单元素集(singleton set)

仅包含一个元素的集合

R10．不能混淆空集与$\{\phi\}$．前者不包含任何元素，后者仅包含1个元素ϕ．我们可以将其理解为：空集是一个空文件夹，但$\{\phi\}$是一个包含空文件夹的文件夹．

D4．集合A为集合B的子集当且仅当集合A中的任一元素都能在集合B中被找到．我们使用记号$A \subseteq B$表示A是B的子集．此时命题$\forall x(x \in A \rightarrow x \in B)$为真．

T1．对于任一集合S，满足$\phi \subseteq S$，亦满足$S \subseteq S$

S12．真子集(proper subset)

当我们想突出集合A是集合B的子集且$A \neq B$时，我们称集合A是集合B的真子集，记作$A \subset B$(注意此处记号有差别)．当$A \subset B$为真时，$A \subseteq B$为真且B中存在不在集合A中的元素．即A是B的真子集时命题$\forall x(x \in A \rightarrow x \in B) \wedge \exists x(x \in B \wedge x \notin A)$为真

D5．假设S是一个集合，则集合S中有n个不同元素$(n \geq 0)$．我们即称S是一个有限集，n为集合S的基数，记为$|S|$

D6．若集合中包含无穷多个元素，则称此集合为无限集

No.

Date

二、the power set 幂集

D₁. 给定一个集合S, S的幂集为S所有子集所构成的集合, 记作$P(S)$

R₁. 空集的幂集为$\{\phi\}$, $\{\phi\}$的幂集为$\{\phi, \{\phi\}\}$

R₂. 若一个集合包含n个元素, 此集合的幂集包含2^n个元素

三、Cartesian products 笛卡儿积

S₁. 有序n元组(ordered n-tuples)

显示有序元素的结构

D₁. 有序n元组$(a_1, a_2 \dots a_n)$为有序的组合, 其中a_1是第一个元素, a_2是第二个元素 … a_n是第n个元素

R₁. 两个有序n元组是等价的当且仅当每一对对应元素均相同. 即$(a_1, a_2 \dots a_n) = (b_1, b_2 \dots b_n)$当且仅当$a_i = b_i$ $(i=1,2 \dots n)$

S₂. 有序对(ordered pairs)

有序二元组称为有序对. 有序对(a,b)和(c,d)等价当且仅当$a=c$, $b=d$

D₂. 假设两集合A.B. A和B的笛卡儿积, 记作$A \times B$, 为所有有序对(a,b)的集合, 其中$a \in A$, $b \in B$. 因此, $A \times B = \{(a,b) | a \in A \land b \in B\}$

S₃. 关系(relation)

笛卡儿积的子集R称为集合A与集合B间的关系. R中元素均为有序对, 每一对中的第一个元素来自集合A, 第二个元素来自集合B

R₂. 笛卡儿积$A \times B$与$B \times A$不等价, 除非$A=\phi$, $B=\phi$, $A \times B=\phi$, 或者$A=B$

R₃. $R \times R$在几何上表示为平面内所有坐标点.

R₄. $|A \times B| = |A| \times |B|$

D₃. 集合$A_1, A_2 \dots A_n$的笛卡儿积记作$A_1 \times A_2 \times \cdots \times A_n$为有序$n$元组所构成的集合, 其中$a_i$为集合$A_i$中的元素$(i=1,2,\dots,n)$

$$A_1 \times A_2 \times \cdots \times A_n = \{(a_1, a_2, \dots a_n) | a_i \in A_i, \text{和 } i=1,2,\dots,n\}$$

四、truth sets of quantifiers 量词的真值集合

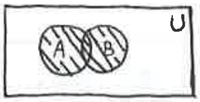
S₁. 真值集合(truth set)

对于给定的谓词P, 与论域D, 我们定义P的真值集合为使$P(x)$为真的D中元素x的集合. $P(x)$的真值集合记作$\{x \in D | P(x)\}$

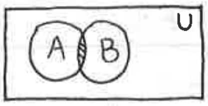
2.2 set operations 集合运算

一、introduction 介绍

D_1. 假设集合A，B，集合A，B的并记为$A\cup B$，为包含A中元素，B中元素的集合. 元素x属于集合A，B的并当且仅当x属于集合A或属于集合B. 即：$A\cup B=\{x\mid x\in A \vee x\in B\}$

维恩图：

D_2. 假设集合A，B，集合A、B的交记为$A\cap B$，为仅包含A，B共有元素的集合. 元素x属于集合A，B的交当且仅当x属于集合A，同时x属于集合B. 即$A\cap B=\{x\mid x\in A \wedge x\in B\}$

维恩图：

D_3. 如果两集合的交为空集，则此两集合是不相交的

S_1. 容斥原理(principle of inclusion-exclusion)

通过分析集合A，B的并的基数$|A\cup B|$与集合A、B的基数$|A|$、$|B|$可得出$|A\cup B|=|A|+|B|-|A\cap B|$. 这个公式的一般化使得我们可以联合任意数目的集合，因而称为容斥原理

D_4. 假设集合A，B，集合A、B的差记为$A-B$，为仅包含A中~~元~~却不在B中元素的集合. A与B的差集亦可称为B对于A的补集. 元素x属于集合A、B的差当且仅当x属于集合A，但不属于集合B. 即$A-B=\{x\mid x\in A \wedge x\notin B\}$

S_2. 补(complement)

一旦全集U被确定，这个集合的补集也被确定下来

D_5. 假设U为全集，集合A的补记作$\overline{A}$，为A对于U的补集. 换言之，集合A的补为$U-A$. 元素x属于集合$\overline{A}$当且仅当$x\notin A$. 即$\overline{A}=\{x\mid x\notin A\}$

R_1. 集合恒等式

等式	名称
$A\cup\emptyset=A$ $A\cap U=A$	恒等律
$A\cup U=U$ $A\cap\emptyset=\emptyset$	支配律
$A\cup A=A$ $A\cap A=A$	幂等律
$\overline{(\overline{A})}=A$	补集律
$A\cup B=B\cup A$ $A\cap B=B\cap A$	交换律
$A\cup(B\cup C)=(A\cup B)\cup C$ $A\cap(B\cap C)=(A\cap B)\cap C$	结合律
$A\cap(B\cup C)=(A\cap B)\cup(A\cap C)$ $A\cup(B\cap C)=(A\cup B)\cap(A\cup C)$	分配律
$\overline{A\cup B}=\overline{A}\cap\overline{B}$ $\overline{A\cap B}=\overline{A}\cup\overline{B}$	德摩根律

等式	名称
$A\cup(A\cap B)=A$ $A\cap(A\cup B)=A$	吸收律
$A\cup\overline{A}=U$ $A\cap\overline{A}=\emptyset$	补律

No.

Date

二、set identities 集合恒等式

R₁. 证明集合恒等式的方法
- ① 证明相互包含
- ② 使用成员表进行证明
- ③ 运用已知集合恒等式证明

S₁. 成员表 (membership tables)

我们可以通过列成员表的方式证明集合恒等式. 我们考虑一个元素可能属于的集合的每一种组合, 并证明在同样的集合组合中的元素属于恒等式两边的集合. 用1表示元素属于一个集合, 用0表示元素不属于一个集合 (与真值表有相似之处)

三、generalized unions and intersections 并集与交集的扩展

D₁、一系列集合的并集为包含所有集合中所有元素的集合

即: $A_1 \cup A_2 \cup \cdots \cup A_n = \bigcup_{i=1}^{n} A_i$

D₂、一系列集合的交集为包含所有集合中共同元素的集合

即: $A_1 \cap A_2 \cap \cdots \cap A_n = \bigcap_{i=1}^{n} A_i$