数理逻辑笔记

陈鸿峥

2020.08*

目录

*Build 20200802

Logic is the study of truth preserving inferences.

数理逻辑的研究分支包括模型论、证明论、集合论和递归论。

句法(syntax)	**语义(semantics)**
形式（可推导关系）	内容（真假）
证明论	模型论
形式语言[1]	解释系统
推断	真值表
没有真和满足，只有代入替换规则	有满足性，没有证明推演
$\Gamma \vdash \phi$	$\Gamma \models \phi$
可靠性(soundness)：左推右$\Gamma \vdash \phi \implies \Gamma \models \phi$	
完备性(completeness)：右推左$\Gamma \models \phi \implies \Gamma \vdash \phi$	

形式系统包括形式语言、公理、推理规则三个部分。一个描述解决多个类似问题域[2]的解决问题框架被称作元(meta)系统，如果代入到具体的问题域则变为对象系统/公理系统。描述问题域的语言称作元语言，而问题域的语言则是对象语言。

1 命题逻辑

定义 1 (命题(proposition)). *命题或声明式句子是指可判断为真或者假的句子。不可被分解的(indecomposable)命题为原子命题。*

关于命题公式的定义在这里不再给出，注意$\rightarrow$是右结合(right-associative)的，如$p \rightarrow q \rightarrow r$等价于$p \rightarrow (q \rightarrow r)$。

1.1 自然推断

定义 2 (自然推断(deduction)). *假设有一系列前提(premise)公式$\phi_1, \phi_2 \ldots, \phi_n$，及结论$\psi$，那么推断过程可记为*

$$\phi_1, \phi_2, \ldots, \phi_n \vdash \psi$$

这一表达式称为一个序列(sequent)，若一个证明可以被找到则称它是有效的(valid)。

[1]只有形式没有内容的语言，即形式语言没有固定的语义，只有跟具体的模型进行了绑定（得到了语义解释）才会有具体语义。

[2]本质上就是模型(model)

推理的基本规则：

- and-introduction ($\wedge i$)：前提与前提为真

$$\frac{\phi \qquad \psi}{\phi \wedge \psi} \wedge i$$

- and-elimination ($\wedge e_i$)：前提与中子成分为真

$$\frac{\phi \wedge \psi}{\phi} \wedge e_1 \qquad \frac{\phi \wedge \psi}{\psi} \wedge e_2$$

- negation-introduction ($\neg\neg i$)

$$\frac{\phi}{\neg\neg\phi}\neg\neg i$$

- negation-elimination ($\neg\neg e$)

$$\frac{\neg\neg\phi}{\phi}\neg\neg e$$

- implication-elimination $\rightarrow e$

$$\frac{\phi \quad \phi \rightarrow \psi}{\psi} \rightarrow e$$

- implies-introduction $\rightarrow i$：其中ϕ为暂时的假设(assumption)，只在盒子内生效

$$\frac{\boxed{\begin{array}{c}\phi \\ \vdots \\ \psi\end{array}}}{\phi \rightarrow \psi} \rightarrow i$$

- or-introduction $\vee i_1, \vee i_2$

$$\frac{\phi}{\phi \vee \psi} \vee i_1 \qquad \frac{\psi}{\phi \vee \psi} \vee i_2$$

- or-elimination $\vee e$

$$\frac{\phi \vee \psi \quad \boxed{\begin{array}{c}\phi \\ \vdots \\ \chi\end{array}} \quad \boxed{\begin{array}{c}\phi \\ \vdots \\ \chi\end{array}}}{\chi} \vee e$$

- bottom-elimination

$$\frac{\bot}{\phi}\bot e$$

- not-elimination

$$\frac{\phi \qquad \neg\phi}{\bot}\neg e$$

- negation

$$\frac{\boxed{\begin{array}{c}\phi\\ \vdots\\ \bot\end{array}}}{\neg\phi}\neg i$$

- 假言易位/拒取式(modus tollens, MT)

$$\frac{\phi\to\psi \quad \neg\psi}{\neg\phi}MT$$

- 反证法(proof by contradition, PBC)：与negation比较

$$\frac{\boxed{\begin{array}{c}\neg\phi\\ \vdots\\ \bot\end{array}}}{\phi}PBC$$

- 排中律(the law of the excluded middle, LEM)

$$\phi\vee\neg\phi\text{必有一个为真}$$

例 1. 证明$p\wedge q, r\vdash q\wedge r$是有效的。

分析. 推理过程如下

1	$p\wedge q$	*premise*	
2	r	*premise*	
3	q	$\wedge e_2$	1
4	$q\wedge r$	$\wedge i$	3, 2

$$\frac{\frac{p\wedge q}{q}\wedge e_2 \quad r}{q\wedge r}\wedge i$$

定义 3 (定理(theorem))**.** 有着合法序列$\vdash\phi$的逻辑公式ϕ称为定理。

定义 4 (矛盾式(contradiction))**.** 有着$\phi\wedge\neg\phi$或$\neg\phi\wedge\phi$形式的公式被称为矛盾式。

定义 5 (可证明等价性(provably equivalent))**.** 令ϕ和ψ为命题逻辑公式，ϕ和ψ是可证明等价的当且仅当序列$\phi\vdash\psi$和$\psi\vdash\phi$都是有效的，或者$\phi\dashv\vdash\psi$

1.2 形式语言

定义 6 (合式公式(well-formed formula, WFF))**.** 一个WFF是

- 一个原子公式（无论是命题常元还是命题变元）

- 形如$(\neg\phi)$的公式，其中ϕ是一个WFF
- 形如$(\phi \vee \psi)$的公式，即由二元连接词连接的两个WFF

WFF可用$BNF(Backus\ Naur\ Form)$定义

$$\phi ::= p \mid \neg\phi \mid \phi \wedge \psi \mid \phi \vee \psi \mid \phi \to \psi \mid (\phi)$$

简而言之，WFF即认为可成为语句的符号串，如$1+1=2$是WFF，而$1=+1\ 2$不是。

例 2. 如下是WFF的一棵语法树*(parse tree)*

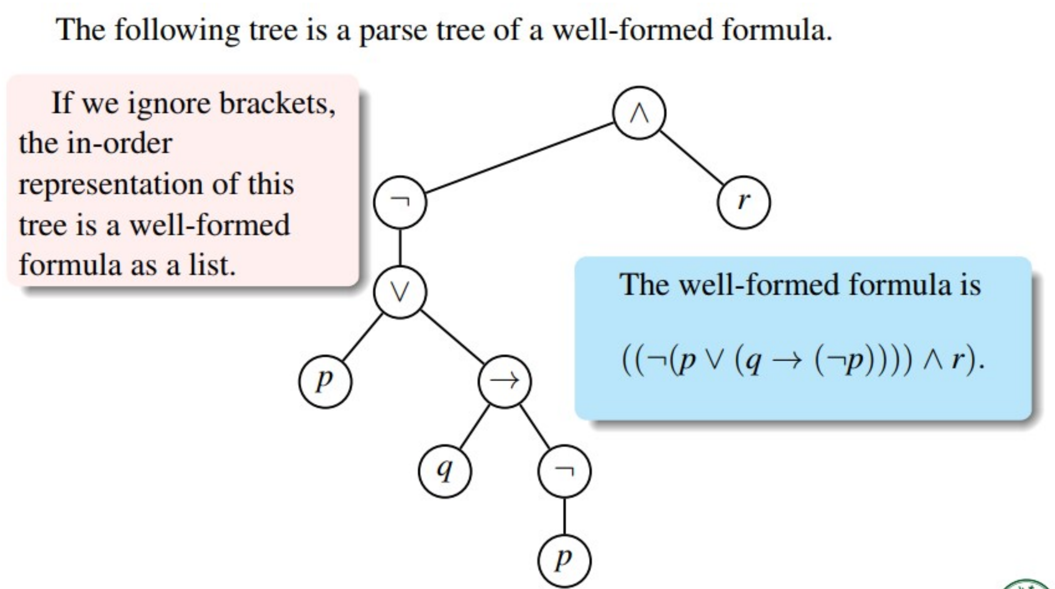


FIG: A parse tree representing a well-formed formula

1.3 语义

定义 7 (模型(model)). 在前提$\phi_1, \phi_2, \ldots, \phi_n$和结论$\psi$上定义另一关系，记作

$$\phi_1, \phi_2, \ldots, \phi_n \models \psi$$

真值包括两个元素T和F，公式ϕ的模型*(model)*或估值*(valuation)*是指对ϕ中的每一原子命题都有一个真值指派*(assignment)*。对$\phi_1, \phi_2, \ldots, \phi_n$的真值指派决定了$\psi$的真值，称为$\psi$的解释*(interpretation)*，可表示为真值表中的一行。

如果对于所有$\phi_1, \phi_2, \ldots, \phi_n$的估值都为$T$，$\psi$也估值为$T$，那么称

$$\phi_1, \phi_2, \ldots, \phi_n \models \psi$$

成立*(hold)*，且称$\models$为语义后承*(semantic entailment)*关系。

定理 1 (可靠性(soundness)). 令$\phi_1, \phi_2, \ldots, \phi_n$和$\psi$都是命题逻辑公式，若$\phi_1, \phi_2, \ldots, \phi_n \vdash \psi$是有效的[3]，那

[3]可以由$\phi_1, \ldots, \phi_n$推出ψ

么$\phi_1, \phi_2, \ldots, \phi_n \models \psi$成立。

定理 2 (完备性(completeness)). 若$\phi_1, \ldots, \phi_n \models \psi$成立，则存在自然推断证明$\phi_1, \ldots, \phi_n \vdash \psi$。

定理 3. 对于命题逻辑来说，完备性和可靠性是等价的。

定义 8 (恒真式(tautology)/矛盾式(contradiction)). 命题逻辑ϕ被称为恒真式当且仅当它在所有估值下都取值为T，也即$\models \phi$。若所有估值均为F，则为矛盾式。

定义 9 (可满足的(satisfiable)). 令ϕ为命题逻辑公式，ϕ是可满足的当且仅当$\neg\phi$不是有效的。

定理 4. 若$\models \eta$成立，则$\vdash \eta$是有效的。换句话说，若η是永真式，则η是定理。

几个常见逻辑符号的区别[4]：

- **语义后承**(semantic consequence)，符号是$\models$`\models`。语义后承在一般情况下是连接一个命题集合和一个命题。如果在任何一种语义赋值$\mathcal{M}$下，只要命题集合Γ中的每一个命题都为真（**真值表**方式），那么ϕ就一定为真，那么我们就说ϕ是Γ的语义后承，记作$\Gamma \models \phi$。
- **句法后承**(syntactic consequence)，符号是$\vdash$`\vdash`。句法后承的用法和语义后承类似，也是连接一个命题集合和一个命题，如$\Gamma \vdash \phi$，表示的是ϕ可以通过**句法证明**的方式从命题集Γ中得出。以Hilbert style的证明为例，这即是说，存在一个命题序列，使得每个前提要么是公理，要么是Γ中的命题，而这个命题序列的最后一项是ϕ。
- **实质蕴含**(material implication / material conditional)，符号是$\to$`\to`，当然在强调和不同蕴涵词对比的时候，我们可能会用箭头表示特殊的蕴涵，而用马蹄符号$\supset$表示实质蕴涵，这个取决于作者)。实质蕴含是一个命题逻辑中的二元算子，连接的是两个命题。在句法系统中，由Hilbert的前两条公理完全刻画，由第三条公理刻画它和否定的关系。在语义系统中，我们说$\mathcal{M} \models \phi \to \psi$当且仅当$\mathcal{M} \models \psi$或者$\mathcal{M} \not\models \phi$。

可以理解为$\vdash$左侧是一些公理(axiom)，右侧是陈述(statement)；而$\to$只是一个连接符，本身并不包含推理的信息。$p \to q$更像是一个数字（如$2+2$），而不是一个判断，它只表达了纯粹的命题内容。要断定它，更应该说“$p \to q$为真”才行。

1.4 范式

定义 10 (语义等价(semantic equivalence)). ϕ和ψ都是命题逻辑的公式，称其等价当且仅当$\phi \models \psi$和$\psi \models \phi$成立，记作$\phi \equiv \psi$，也等价于$\models (\phi \to \psi) \land (\psi \to \phi)$成立，也即同时取真或同时取假。

定义 11 (合取范式(conjunction normal form, CNF)). *BNF*定义如下：

---

[4]参考以下资料：

- https://math.stackexchange.com/questions/2903877/to-vs-vdash-in-logic
- 逻辑学中，前提为假而命题为真的推论如何解释？ - 罗心澄的回答 - 知乎 https://www.zhihu.com/question/21020308/answer/16917222
- 数理逻辑 $\implies$ ，$\vdash$ - 这两个符号有什么区别？ - 罗心澄的回答 - 知乎 https://www.zhihu.com/question/21191299/answer/17469774
- 逻辑学蕴涵命题中的$\to$和数学中的 $\implies$ 有什么区别和共同点？ - 罗心澄的回答 - 知乎 https://www.zhihu.com/question/276859264/answer/459951353

- 文字(*literal*)：$L ::= p \mid \neg p$
- 句子(*clause*)：$D ::= L \mid L \vee D$
- 公式(*formula*)：$C ::= D \mid (D) \mid D \wedge C$

例子如

$$(p \vee r) \wedge (\neg p \vee r) \wedge (p \vee \neg r)$$

定义 12 (霍尔公式(Horn formula))**.** 若命题逻辑公式ϕ能用下面的语法，表示成H的一个示例

$$P ::= \bot \mid \top \mid p \qquad A ::= P \mid P \wedge A \qquad C ::= A \to P \qquad H ::= C \mid C \wedge H$$

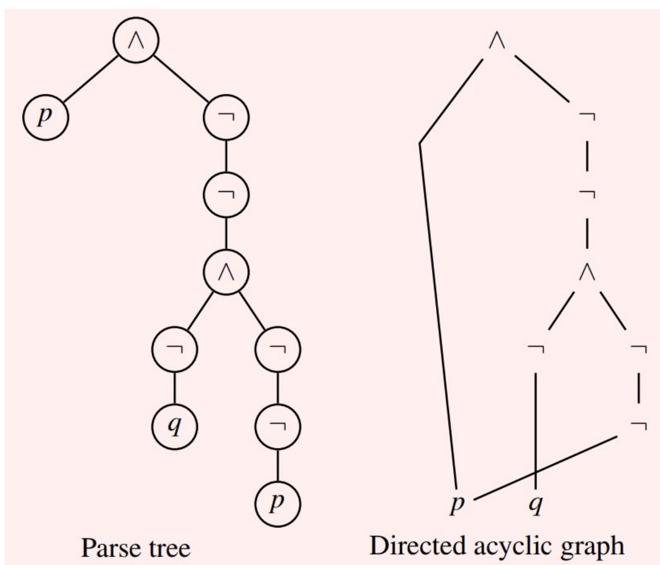
则称C的每个实例为霍尔子句(*clause*)。

1.5 SAT求解器

线性求解器只接受以下几种形式的公式

$$\phi ::= p \mid (\neg \phi) \mid (\phi \wedge \phi)$$

例 3. $\phi = p \wedge \neg(q \vee \neg p)$，计算$T(\phi) = p \wedge \neg\neg(\neg q \wedge \neg\neg p)$，则有语法树和$DAG$如下



2 谓词逻辑

谓词逻辑(predicate)或被称为一阶逻辑，提供了更强的语言表达能力。

2.1 形式语言

定义 13 (项(item))**.** 项的定义如下：

- 每一个变量都是项

- 若$c \in \mathcal{F}$是一个空函数，则c是项（常数）
- 若$t_1, t_2, \ldots, t_n$都是项，且$f \in \mathcal{F}$有$n > 0$个元*(arity)*，则$f(t_1, t_2, \ldots, t_n)$是一个项

用*BNF*写即

$$t ::= x \mid c \mid f(t_1, t_2, \ldots, t_n)$$

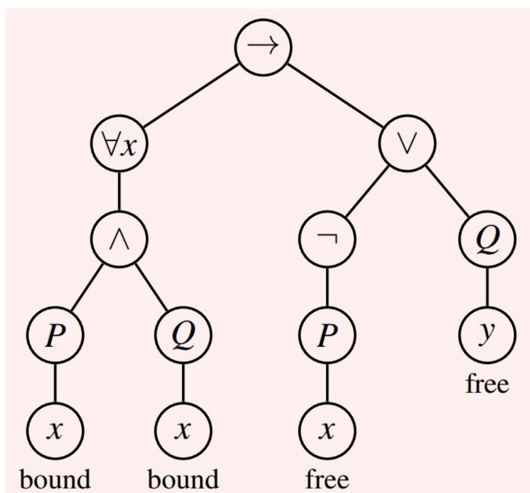
定义 14 (公式(formula)). *BNF*定义如下

$$\phi ::= P(t_1, t_2, \ldots, t_n) \mid (\neg\phi) \mid (\phi \wedge \psi) \mid (\phi \vee \psi) \mid (\phi \to \psi) \mid (\forall x \phi) \mid (\exists x \phi)$$

运算符优先级如下

- $\neg, \forall, \exists$
- $\vee, \wedge$
- $\to$

定义 15 (自由(free)/约束(bound)变量). 语法树叶结点往上不会经过$\forall x$或$\exists x$结点，则为自由变量。



定义 16 (替代(substituion)). 给定变量x、项t和公式ϕ，定义$\phi[t/x]$为将ϕ中所有**自由的**x用t代替

2.2 证明论

- equality $(= i)$

$$\frac{}{t = t} = i$$

- substitution $(= e)$

$$\frac{t_1 = t_2 \qquad \phi[t_1/x]}{\phi[t_2/x]} = e$$

- for-eliminating $(\forall x\ e)$

$$\frac{\forall x \phi}{\phi[t/x]} \forall x\ e$$

- for-introduction ($\forall x\ i$)

$$\dfrac{\boxed{\begin{array}{c} x_0(\text{fresh/dummy var}) \\ \vdots \\ \phi[x_0/x] \end{array}}}{\forall x \phi}\forall x \quad i$$

- exists-introduction ($\exists x\ i$)

$$\dfrac{\phi[t/x]}{\exists x \phi}\exists x\ i$$

- exists-elimination ($\exists e$)

$$\dfrac{\exists x\phi \quad \boxed{\begin{array}{c} x_0 \quad \phi[x_0/x] \\ \vdots \\ \chi \end{array}}}{\chi}\exists e$$

例 4. 证明$\forall x(P(x) \to Q(x)), \forall x P(x) \vdash \forall x Q(x)$

分析. 推理过程如下

1	$\forall x(P(x) \to Q(x))$	*premise*
2	$\forall x P(x)$	*premise*
3	$P(x_0) \to Q(x_0)$	$\forall x \quad e \quad 1$
4	$P(x_0)$	$\forall x \quad e \quad 2$
5	$Q(x_0)$	$\to e \quad 3,4$
6	$\forall x Q(x)$	$\forall x \quad i \quad 3-5$

定理 5 (等价性(equivalence))**.** 令ϕ和ψ都为谓词逻辑的公式，有以下等价性

$$\neg\forall x\phi \dashv\vdash \exists x\neg\phi, \qquad \neg\exists x\phi \dashv\vdash \forall x\neg\phi$$

对于自然语言的形式化，有几点需要注意[5]：

1. 在自然语言命题中，如果是对个体类的全体判断具有某个性质，而且有提取对应个体类的谓词，通常这个谓词与对应性质的谓词之间的逻辑联结关系是**蕴涵**。例如“老师上课很认真”中的“x是老师”的$P(x)$和“x上课很认真”的$Q(x)$之间是蕴涵关系“对任意的x，如果x是老师，则x上课很认真”，符号化为$\forall x(P(x) \to Q(x))$。也许部分同学觉得应该符号化为$\forall x(P(x) \land Q(x))$，但这是错误的，如果论域是所有的人构成的集合，并且存在不是老师的人，那么在这个论域中$\forall x(P(x) \land Q(x))$的真值为假，因为对于不是老师的元素$a$，$P(a) \land Q(a)$为假，而原来的命题则无论论域是什么，只是对老师进行判断，其真值应该只与老师上课是否认真有关，而不应该由人是否老师确定。
2. 对偶地，如果是对各类的一个或多个不确定的个体判断具有某个性质，而且有提取对应个体类的谓词，通常这个谓词与对应性质的谓词之间的逻辑联结关系是**合取**。例如命题“有的学生在讲小话”，

[5]摘自中山大学周晓聪老师的《离散数学》课程

如果设$S(x)$表示“x是学生”，$T(x)$表示“x在讲小话”，则整个命题的相当于：“存在x，x是学生，而且x在讲小话”，符号化为$\exists x(S(x)\land T(x))$。注意不能符号化为$\exists x(S(x)\to T(x))$，因为当论域中存在不是学生的元素$a$，即$S(a)$为假，那么根据蕴涵联结词的真值计算，$S(a)\to T(a)$总为真，从而公式$\exists x(S(x)\to T(x))$的真值就是真，而原命题的真值显然只应该取决于是否有学生在讲小话，而不应该由是否存在不是学生的元素决定。

类似地，对于数学上常用的带有受限论域的量词公式，如$\forall x<0(x^2>0)$和$\exists z>0(z^2=2)$要记住其含义分别$\forall x(x<0\to x^2>0)$和$\exists z(z>0\land z^2=2)$，使用全称量词约束就是蕴涵的关系，而使用存在量词就是合取的关系！

2.3 语义

记Γ为公式$\phi_1,\ldots,\phi_n$，要证明$\Gamma\vdash\psi$是合法的，则只需从Γ中提供ψ的证明。而如果要证明Γ推不出ψ，从**证明论**(proof theory)的角度是困难的。

但从语义(semantics)的角度，只需找到一个模型(model)满足所有ϕ_i都为真，而ψ为假，即可得到Γ推不出ψ；相反地，要证明Γ推出ψ则是困难的，因为对于谓词逻辑来说有无穷多种估值/模型，只有全部验证了才能得知$\Gamma\models\psi$（ψ被Γ语义蕴含entail）。

因此证明论和模型论两者都是重要的。

定义 17 (模型(model)). *令$\mathcal{F}$为函数符号的集合，$\mathcal{P}$为谓词符号的集合*[6]*。关于$(\mathcal{F},\mathcal{P})$的模型$\mathcal{M}$包含以下数据：*

- *非空集合A：具体值的全集*
- *对于每一空函数符号$f\in\mathcal{F}$, $f^{\mathcal{M}}\in A$*
- *对于每一$f\in\mathcal{F}$且元$n>0$，具体函数$f^{\mathcal{M}}:A^n\to A$*
- *对于每一$P\in\mathcal{P}$且元$n>0$，子集$P^{\mathcal{M}}\subset A^n$*

*也就是说f和P仅仅是**抽象的符号**，而$f^{\mathcal{M}}$和$P^{\mathcal{M}}$为**具体的函数/元素**。*
也可以说模型给出了一个解释(interpretation)。

例 5. *令$\mathcal{F}\overset{def}{=\!=}\{i\}$，$\mathcal{P}\overset{def}{=\!=}\{R,F\}$，其中$i$是一个常数，$F$是一元谓词符号，$R$是二元谓词符号。一个模型可以是*

$$\begin{aligned}A&\overset{def}{=\!=}\{a,b,c\}\\ i^{\mathcal{M}}&\overset{def}{=\!=}a\\ R^{\mathcal{M}}&\overset{def}{=\!=}\{(a,a),(a,b),(a,c),(b,c),(c,c)\}\\ F^{\mathcal{M}}&\overset{def}{=\!=}\{b,c\}\end{aligned}$$

那么有$\exists yR(i,y)$为真，$\neg F(i)$为真。

上面给出了模型的定义，并且使得我们可以直接从$(\mathcal{F},\mathcal{P})$中计算出真值，但我们仍需讨论如何处理全称量词$\forall x\phi$及特称量词$\exists x\phi$，需要检查$\phi$是否对于所有模型中的$a$都成立。尽管我们可以用$\phi[a/x]$表示，但是$\phi[a/x]$并不是一个逻辑公式，因为$a$不是一个项(term)而是模型中的一个元素。

[6]函数符号是一个算子，作用在项上并生成一个新的项/实体(object)，比如+和×；而谓词符号也是一个算子，作用在项上并生成一个谓词/宣称(claim)，比如<和>。

因此需要限定公式是关于一个环境的(relative to an environment)。

定义 18 (环境(environment)/查找表(look-up table))**.** 对于全集A具体值的环境是一个函数$l : var \to A$，定义$l[x \mapsto a]$为从x映射到a的查找表。

定义 19 (满足关系(satisfaction))**.** 给定$\mathcal{M}(\mathcal{F}, \mathcal{P})$及环境$l$，定义满足关系$\mathcal{M} \models_l \phi$。若$\mathcal{M} \models_l$成立*(hold)*，则称$\phi$在关于环境$l$的模型$\mathcal{M}$中计算为$T$。

例 6. 令$F \overset{def}{=\!=} \{alma\}, P \overset{def}{=\!=} \{loves\}$，其中$alma$是常数，$loves(\cdot,\cdot)$是谓词符号。模型$\mathcal{M}$包含集合

$$A \overset{def}{=\!=} \{a, b, c\}, alma^{\mathcal{M}} \overset{def}{=\!=} a, loves^{\mathcal{M}} \overset{def}{=\!=} \{(a, a), (b, a), (c, a)\}$$

检查模型$\mathcal{M}$是否满足

None of Alma's lovers' lovers love her.

$$\phi : \forall x \forall y (loves(x, alma) \wedge loves(y, x) \to \neg loves(y, alma))$$

令$x \mapsto a$及$y \mapsto b$，又$alma^{\mathcal{M}} \overset{def}{=\!=} a$，得到

$$loves(a, a) \wedge loves(b, a) \to \neg loves(b, alma)$$

不成立，故$\mathcal{M} \not\models \phi$

在命题逻辑中，$\phi_1, \ldots, \phi_n \models \psi$[7]当且仅当$\phi_1, \ldots, \phi_n$都估值为$T$，同时$\psi$也估值为$T$。

定义 20. 令Γ为谓词逻辑的公式集合，ψ为谓词逻辑的公式

1. 语义蕴含$\Gamma \models \psi$成立当且仅当对于**所有**的模型$\mathcal{M}$和查找表l，只要$\forall \phi \in \Gamma : \mathcal{M} \models_l \phi$成立，$\mathcal{M} \models_l \psi$也成立
2. ψ是可满足的*(satisfiable)*当且仅当**存在**模型$\mathcal{M}$和环境l使得$\mathcal{M} \models_l \psi$成立
3. ψ是合法的*(valid)*当且仅当$\mathcal{M} \models_l \psi$对于**所有**模型$\mathcal{M}$和环境$l$都成立

例 7. 考虑以下语义蕴含关系

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

令$\mathcal{M}$为满足$\forall x (P(x) \to Q(x))$的模型，需要证明$\mathcal{M}$也满足$\forall x P(x) \to \forall x Q(x)$

- 若不是每个$\mathcal{M}$中的元素都满足P，则前件为假，显然成立
- 若每个$\mathcal{M}$中的元素都满足P，则每一元素都满足Q，因为$\mathcal{M}$满足$\forall x (P(x) \to Q(x))$

进而$\mathcal{M} \models \forall x P(x) \to \forall x Q(x)$

例 8. 考虑以下语义蕴含关系

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

[7]$\models$代表语义蕴含(semantic entailment)

令$A \xlongequal{def} \{a,b\}, P^{\mathcal{M}} \xlongequal{def} \{a\}, Q^{\mathcal{M}} \xlongequal{def} \{b\}$，则$\mathcal{M} \models \forall x P(x) \to \forall x Q(x)$，但$\mathcal{M} \not\models \forall x (P(x) \to Q(x))$

3 模型验证

3.1 线性时序逻辑

定义 21 (线性时序逻辑(linear-time temporal logic, LTL)). *BNF*定义如下

$$\phi ::= \top \mid \bot \mid p \mid (\neg\phi) \mid (\phi \wedge \phi) \mid (\phi \vee \phi) \mid (\phi \to \phi) \mid (X\phi) \mid (F\phi) \mid (G\phi) \mid (\phi U \phi) \mid (\phi W \phi) \mid (\phi R \phi)$$

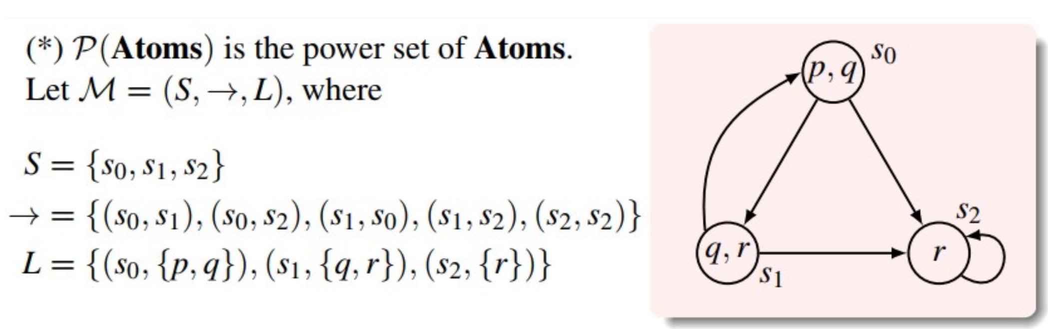
其中X, F, G, U, R, W都称为时序连接词*(temporal connectives)*

- X：*neXt state*
- F：*some Future state*（存在）
- G：*all future states (Globally)*（任意）
- U：*Until*（二元）
- R：*Release*（二元）
- W：*Weak-until*（二元）

运算符优先级

- 一元连接词：$\neg, X, F, G$
- U, R, W
- $\wedge, \vee$
- $\to$

定义 22 (转移系统(transition system)). 一个转移系统$\mathcal{M} = (S, \to, L)$是状态集合$S$（静态结构），转移关系$\to$（动态结构），使得$\forall s \in S, \exists s' \in S$：$s \to s'$，且有标注函数$L: S \to \mathcal{P}(Atoms)$，其中$\mathcal{P}(Atoms)$为原子描述的幂集（实际上$L$就是给所有命题原子做真值指派）。转移系统也可以被称作模型*(model)*。



定义 23 (路径(path)). 路径是一个无穷序列$s_1, s_2, \ldots \in S$使得$s_i \to s_{i+1}, \forall i \geq 1$。令$\pi^i = s_i \to s_{i+1} \to \cdots$为从状态$s_i$开始的路径。路径$\pi = s_1 \to s_2 \to \cdots$满足$LTL$公式定义在满足关系$\models$上：

1. $\pi \models \top$

2. $\pi \not\models \perp$

3. $\pi \models p$ *iff* $p \in L(s_1)$

4. $\pi \models \neg\phi$ *iff* $\pi \not\models \phi$

5. $\pi \models \phi \wedge \psi$ *iff* $\pi \models \phi$ *and* $\pi \models \psi$

6. $\pi \models \phi \vee \psi$ *iff* $\pi \models \phi$ *or* $\pi \models \psi$

7. $\pi \models \phi \rightarrow \psi$ *iff* $\pi \models \psi$ *when* $\pi \models \phi$

8. $\pi \models X\phi$ *iff* $\pi^2 \models \phi$

9. $\pi \models G\phi$ *iff* $\forall i \geq 1:\ \pi^i \models \phi$

10. $\pi \models F\phi$ *iff* $\exists i \geq 1:\ \pi^i \models \phi$

11. $\pi \models \phi U \psi$ *iff* $\exists i \geq 1:\ \pi^i \models \psi$ *and* $\forall j = 1, \ldots, i-1:\ \pi^j \models \phi$

12. $\pi \models \phi W \psi$ *iff* ***either*** $\exists i \geq 1:\ \pi^i \models \psi$ *and* $\forall j = 1, \ldots, i-1:\ \pi^j \models \phi$ ***or*** $\forall k \geq 1:\ \pi^k \models \phi$

13. $\pi \models \phi R \psi$ *iff* ***either*** $\exists i \geq 1:\ \pi^i \models \phi$ *and* $\forall j = 1, \ldots, i:\ \pi^j \models \psi$ ***or*** $\forall k \geq 1:\ \pi^k \models \psi$

记$\mathcal{M}, s \models \phi$表示对于所有$\mathcal{M}$开始于$s$的执行路径$\pi$，都有$\pi \models \phi$，或可简写为$s \models \phi$

注意有以下恒等式

$$\neg G\phi \equiv F\neg\phi$$
$$\neg F\phi \equiv G\neg\phi$$
$$\neg X\phi \equiv X\neg\phi$$
$$\phi W \psi \equiv \phi U \psi \vee G\phi$$
$$\phi R \psi \equiv \neg(\neg\phi U \neg\psi)$$
$$F(\phi \vee \psi) \equiv F\phi \vee F\psi$$
$$G(\phi \wedge \psi) \equiv G\phi \wedge G\psi$$
$$F\phi \equiv \top U \phi$$
$$G\phi \equiv \perp R \phi$$
$$\phi U \psi \equiv \phi W \psi \wedge F\psi$$
$$\phi W \psi \equiv \phi U \psi \vee G\phi$$
$$\phi W \psi \equiv \psi R(\phi \vee \psi)$$
$$\phi R \psi \equiv \psi W(\phi \wedge \psi)$$

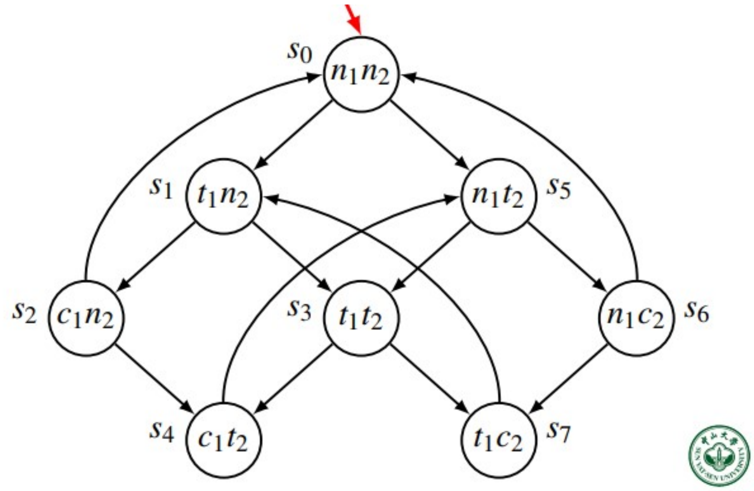
例 9. 对于上图的例子，有以下式子成立

- $s_0 \models p \wedge q$ *(3,5)*
- $s_0 \models Xr$，因$r \in L(s_1), L(s_2)$
- $s_0 \models G\neg(p \wedge r)$，因所有从$s_0$开始的路径都满足$G\neg(p \wedge r)$，也即满足$\neg(p \wedge r)$

3.2 模型检查

下面是一个互斥锁的例子，n为非临界区状态，t为尝试进入临界区，c为临界区状态。考虑2个进程，每个进程的转移都是$n \to t \to c \to n \to \cdots$。

No.	P_1	P_2
0	n_1	n_2
1	n_1	t_2
2	n_1	c_2
3	t_1	n_2
4	t_1	t_2
5	t_1	c_2
6	c_1	n_2
7	c_1	t_2
8	c_1	c_2



有以下性质：

- 安全性(safety)：$G\neg(c_1 \wedge c_2)$在每个状态都被满足
- 活性(liveness)：$G(t_1 \to Fc_1)$，对于路径$s_0 \to s_1 \to s_3 \to s_7 \to s_1 \to s_3 \to \cdots$就不满足
- 非阻塞(non-blocking)：对于任意状态满足n_1，有后继状态满足t_1，无法用LTL表达
- 非严格序列(strict sequencing)

3.3 计算树逻辑

定义 24 (计算树逻辑(computation tree logic, CTL)). *除了LTL有的U, F, G, X，CTL还有A和E表示***所有路径***(All path)和***存在一条路径***(Exist a path)。BNF定义如下*

$$\phi ::= \top \mid \bot \mid p \mid (\neg\phi) \mid (\phi \wedge \phi) \mid (\phi \vee \phi) \mid (\phi \to \phi) \mid (AX\phi) \mid (EX\phi) \mid$$
$$(AF\phi) \mid (EF\phi) \mid (AG\phi) \mid (EG\phi) \mid A[\phi U \phi] \mid E[\phi U \phi]$$

运算符优先级

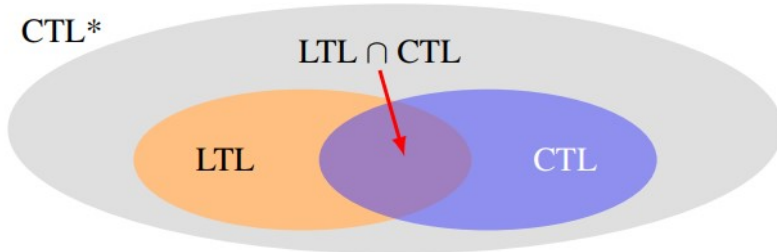
- $\neg, AG, EG, AF, EF, AX, EX$
- $\wedge, \vee$
- $\to, AU, EU$

定义 25 (CTL*). *ϕ为状态公式（在状态上求值），α为路径公式（在路径上求值）*

$$\phi ::= \top \mid p \mid (\neg\phi) \mid (\phi \wedge \phi) \mid A[\alpha] \mid E[\alpha]$$
$$\alpha ::= \phi \mid (\neg\alpha) \mid (\alpha \wedge \alpha) \mid (\alpha U \alpha) \mid (G\alpha) \mid (F\alpha) \mid (X\alpha)$$

- LTL is a subset of CTL*
  The LTL formula α is equivalent to the CTL* formula $A[\alpha]$.
- CTL is a subset of CTL*
  The CTL formula is the fragment of CTL* in which we restrict the form of path formulas.

$$\alpha ::= (\alpha\,\mathrm{U}\,\alpha) \mid (\mathrm{G}\,\alpha) \mid (\mathrm{F}\,\alpha) \mid (\mathrm{X}\,\alpha)$$



The expressive powers of LTL, CTL and CTL*

4 程序验证

4.1 框架

定义 26 (霍尔三元组(Hoare triple))**.**

$$(\!|\phi|\!)P(\!|\psi|\!)$$

代表若程序P在满足ϕ的状态下运行，则执行完P的状态会满足ψ。 ϕ称为先验条件*(precondition)*，ψ称为后验条件*(postcondition)*。存储变量x记为$l(x)$。有函数符号的公式ϕ：$-$*(unary)*, $+$, $-$, $*$*(binary)*, $<$, $=$

例 10. 若输入x是正数，则求出一个数字它的平方小于x。
记程序为P，则*Hoare*三元组为

$$(\!|x > 0|\!)P(\!|y \cdot y < x|\!)$$

一个可行的程序P可以是

```
y = 0;
while (y * y < x) { y = y + 1; }
y = y - 1;
```

定义 27 (正确性)**.** 若对于**所有**满足先验条件ϕ的状态，**只要**P**需要能停止*(terminate)***，经过P的执行，都满足后验条件ψ，则$(\!|\phi|\!)P(\!|\psi|\!)$满足部分正确性*(partial correctness)*。若$\models_{par} (\!|\phi|\!)P(\!|\psi|\!)$成立，则称$\models_{par}$为部分正确性关系。全部正确性*(total correctness)*则是保证了P一定会停止。

例 11. 考虑下面求阶乘的程序*Fac1*:

```
y = 1;
z = 0;
```

```
while (z != x) {
  z = z + 1;
  y = y * z;
}
```

- $\models_{tot} (\!|x \geq 0|\!) Fac1 (\!|y = x!|\!)$成立，只要$x \geq 0$，*Fac1*一定会停止，并且有结果$y = x!$
- $\models_{tot} (\!|\top|\!) Fac1 (\!|y = x!|\!)$不保证成立，因为*Fac1*对于$x$的负数值不会停止
- $\models_{par} (\!|x \geq 0|\!) Fac1 (\!|y = x!|\!)$和$\models_{par} (\!|\top|\!) Fac1 (\!|y = x!|\!)$成立

定义 28 (逻辑变量(logical variable)). 逻辑变量在ϕ或ψ是自由的，且不出现在P中

例 12. $(\!|x = x_0 \wedge x \geq 0|\!) Sum (\!|z = x_0(x_0+1)/2|\!)$

```
z = 0;
while (x != 0) {
  z = z + x;
  x = x - 1;
}
```

4.2 证明论

- Composition
$$\frac{(\!|\phi|\!)\ C_1\ (\!|\eta|\!) \qquad (\!|\eta|\!)\ C_2\ (\!|\psi|\!)}{(\!|\phi|\!)\ C_1; C_2\ (\!|\psi|\!)}$$
- Assignment
$$\frac{}{(\!|\psi[E/x]|\!)\ x = E\ (\!|\psi|\!)}$$
注意这条公式相当tricky，是在先验条件中将x换为E（即恢复$x = E$的赋值），在具体证明中往往是反过来用
- If-statement
$$\frac{(\!|\phi \wedge B|\!)\ C_1\ (\!|\psi|\!) \qquad (\!|\phi \wedge \neg B|\!)\ C_2\ (\!|\psi|\!)}{(\!|\phi|\!)\ \text{if } B\{C_1\} \text{ else } \{C_2\}\ (\!|\psi|\!)}$$
- Partial-while
$$\frac{(\!|\psi \wedge B|\!)\ C\ (\!|\psi|\!)}{(\!|\psi|\!)\ \text{while } B\{C\}\ (\!|\psi \wedge \neg B|\!)}$$
- Implied
$$\frac{\vdash_{AR} \phi' \to \phi \qquad (\!|\phi|\!)\ C\ (\!|\psi|\!) \qquad \vdash_{AR} \psi \to \psi'}{(\!|\phi'|\!)\ C\ (\!|\psi'|\!)}$$
基本谓词逻辑演算及算术表达都满足

例 13 (表格证明(proof tableaux)). $\vdash_{par} (\!|y < 3|\!)\ y = y + 1\ (\!|y < 4|\!)$

$$
\begin{array}{ll}
(\!|y < 3|\!) & \\
(\!|y + 1 < 4|\!) & \textit{Implied} \\
y = y + 1 & \\
(\!|y < 4|\!) & Assignment
\end{array}
$$

$(\!|\phi|\!)$ if $(B)\{C_1\}$ else $\{C_2\}$ $(\!|\psi|\!)$

- 将ψ反推C_1，结果记为ϕ_1
- 将ψ反推C_2，结果记为ϕ_2
- 令$\phi = (B \to \phi_1) \wedge (\neg B \to \phi_2)$

例 14. 考虑程序$Succ$

```
a = x + 1;
if (a-1 == 0) { y = 1; } else { y = a; }
```

证明$\vdash_{par} (\!|\top|\!)\ Succ\ (\!|y = x + 1|\!)$合法

分析. 可以自底向上分析

```
  (|⊤|)
  (|(x + 1 − 1 = 0 → 1 = x + 1) ∧ (¬(x + 1 − 1 = 0) → x + 1 = x + 1)|) Implied
a = x + 1;
  (|(a − 1 = 0 → 1 = x + 1) ∧ (¬(a − 1 = 0) → a = x + 1)|) Assignment
if (a - 1 == 0) {
    (|1 = x + 1|) If-statements
  y = 1;
    (|y = x + 1|) Assignment
} else {
    (|a = x + 1|) If-statements
  y = a;
    (|y = x + 1|) Assignment
}
    (|y = x + 1|) If-statements
```

$(\!|\phi|\!)$ while $B\{C\}$ $(\!|\psi|\!)$

需要找到合适的不变量(invariant)η使得

- $\models_{AR} \phi \to \eta$
- $\models_{AR} \eta \wedge \neg B \to \psi$
- $\models_{par} (\!|\eta|\!)$ while $(B)\{C\}(\!|\eta \wedge \neg B|\!)$

例 15. 考虑程序$Fac1$

```
y = 1;
z = 0;
while (z != x) { // L1
  z = z + 1;
  y = y * z;
} // L2
```

证明

$$\models_{AR} (y = 1 \land z = 0) \to (y = z!) \qquad \models_{AR} (y = z! \land x = z) \to (y = x!)$$

分析. 自底向上分析

```
  (|⊤|)
  (|1 = 0!|) Implied
y = 1;
  (|y = 0!|) Assignment
z = 0;
  (|y = z!|) Assignment
while (z != x) {
    (|y = z! ∧ z ≠ x|) Invariant Hyp.  ∧ guard
    (|y · (z + 1) = (z + 1)!|) Implied
  z = z + 1;
    (|y · z = z!|) Assignment
  y = y * z;
    (|y = z!|) Assignment
}
  (|y = z! ∧ ¬(z ≠ x)|) Partial-while
  (|y = x!|) Implied
```

5 模态逻辑

定义 29 (模态逻辑(modal logic))**.** *BNF*定义如下

$$\phi ::= \top \mid \bot \mid p \mid (\neg\phi) \mid (\phi \land \phi) \mid (\phi \lor \phi) \mid (\phi \to \phi) \mid (\phi \leftrightarrow \phi) \mid (\Box\phi) \mid (\Diamond\phi)$$

其中$\Box$为一定*(necessarily)*，$\Diamond$为可能*(possibly)*。优先级顺序

- $\neg, \Box, \Diamond$
- $\lor, \land$
- $\to, \leftrightarrow$

定义 30 (模型(model))**.** 模态逻辑的模型$\mathcal{M}$由以下三个成分定义：

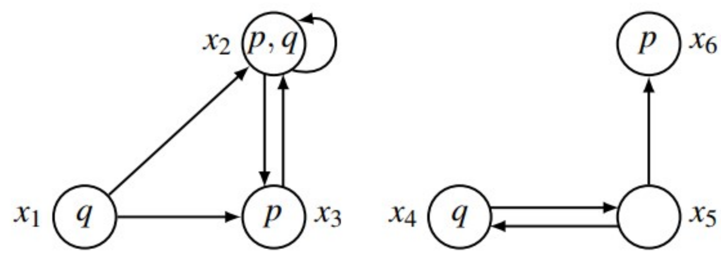
- 世界*(world)*元素W

- 定义在W上的可访问*(accessibility)*关系$R \subset W \times W$
- 标记函数*(labeling)*$L : W \to \mathcal{P}(Atoms)$

这些模型称为*Kripke*模型

- $W = \{x_1, x_2, x_3, x_4, x_5, x_6\}$
- $R = \{(x_1, x_2), (x_1, x_3), (x_2, x_2), (x_2, x_3), (x_3, x_2), (x_4, x_5), (x_5, x_4), (x_5, x_6)\}$
- the labelling function $L : W \to \mathcal{P}(Atoms)$ is as follows:

x	x_1	x_2	x_3	x_4	x_5	x_6
$L(x)$	$\{q\}$	$\{p, q\}$	$\{p\}$	$\{q\}$	$\emptyset$	$\{p\}$



A Kripke model

定义 31. 令$\mathcal{M} = (W, R, L)$为基本的模态逻辑，$x \in W$和ϕ是公式，满足性*(satisfaction)*关系$x \Vdash \phi$为在ϕ上的结构推断*(structural induction)*:

$$
\begin{array}{lll}
x & \Vdash \top & \\
x & \nVdash \bot & \\
x & \Vdash p & \textit{iff } p \in L(x) \\
x & \Vdash \neg\phi & \textit{iff } x \nVdash \phi \\
x & \Vdash \phi \wedge \psi & \textit{iff } x \Vdash \phi \textit{ and } x \Vdash \psi \\
x & \Vdash \phi \vee \psi & \textit{iff } x \Vdash \phi \textit{ or } x \Vdash \psi \\
x & \Vdash \phi \to \psi & \textit{iff } x \Vdash \psi \textit{ whenever } x \Vdash \phi \\
x & \Vdash \phi \leftrightarrow \psi & \textit{iff } (x \Vdash \phi \textit{ iff } x \Vdash \psi) \\
x & \Vdash \Box\psi & \textit{iff } \forall y \in W, R(x, y) :\ y \Vdash \psi \\
x & \Vdash \Diamond\psi & \textit{iff } \exists y \in W, R(x, y) :\ y \Vdash \psi
\end{array}
$$

例 16. 比如上图的例子有

- $x_1 \Vdash q$，因$q \in L(x_1)$
- $x_1 \Vdash \Diamond q$，因$R(x_1) = \{x_2, x_3\}, x_2 \in R(x_1), q \in L(x_2)$
- $x_1 \nVdash \Box q$，因$R(x_1) = \{x_2, x_3\}, x_3 \in R(x_1), q \notin L(x_3)$

De Morgan定律

$$\neg\Box\phi \equiv \Diamond\neg\phi \qquad \neg\Diamond\phi \equiv \Box\neg\phi$$

分配律

$$\Box(\phi \wedge \psi) \equiv \Box\phi \wedge \Box\psi \qquad \Diamond(\phi \vee \psi) \equiv \Diamond\phi \vee \Diamond\psi$$

K模式(scheme)

$$\Box(\phi \to \psi) \wedge \Box\phi \to \Box\psi$$

例 17. 证明$\neg\Box\phi \equiv \Diamond\neg\phi$

分析. 假设x是模型$\mathcal{M} = (W, R, L)$的一个世界，希望找到$x \Vdash \neg\Box\phi \leftrightarrow \Diamond\neg\phi$

$$\begin{aligned} & x \Vdash \neg\Box\phi \\ \leftrightarrow & \textit{it is not the case that } x \Vdash \Box\phi \\ \leftrightarrow & \textit{it is not the case that } \forall y \in R(x), y \Vdash \phi \\ \leftrightarrow & \exists y \in R(x) \textit{ and not } y \Vdash \phi \\ \leftrightarrow & \exists y \in R(x) \textit{ and } y \Vdash \neg\phi \\ \leftrightarrow & x \Vdash \Diamond\neg\phi \end{aligned}$$

6 二元决策图

$\cdot$为与，$+$为或，$\oplus$为异或。

布尔函数用真值表表示，如$f(x,y) \xlongequal{\text{def}} \overline{x+y}$。

x	y	$f(x,y)$
1	1	0
1	0	1
0	1	0
0	0	1

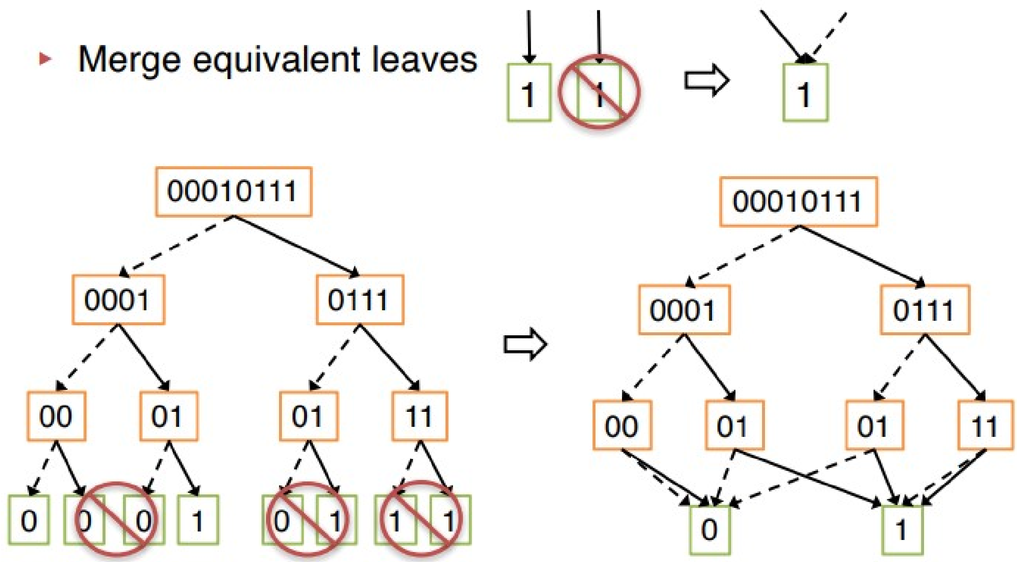
二元决策图(Binary Decision Diagram, BDD)

- 非终端结点标号为布尔变量
- 终端结点（叶子）标号为0或1
- 每一个非终端结点有两条边，一条虚边(dashed)指向0，一条实边(solid)指向1

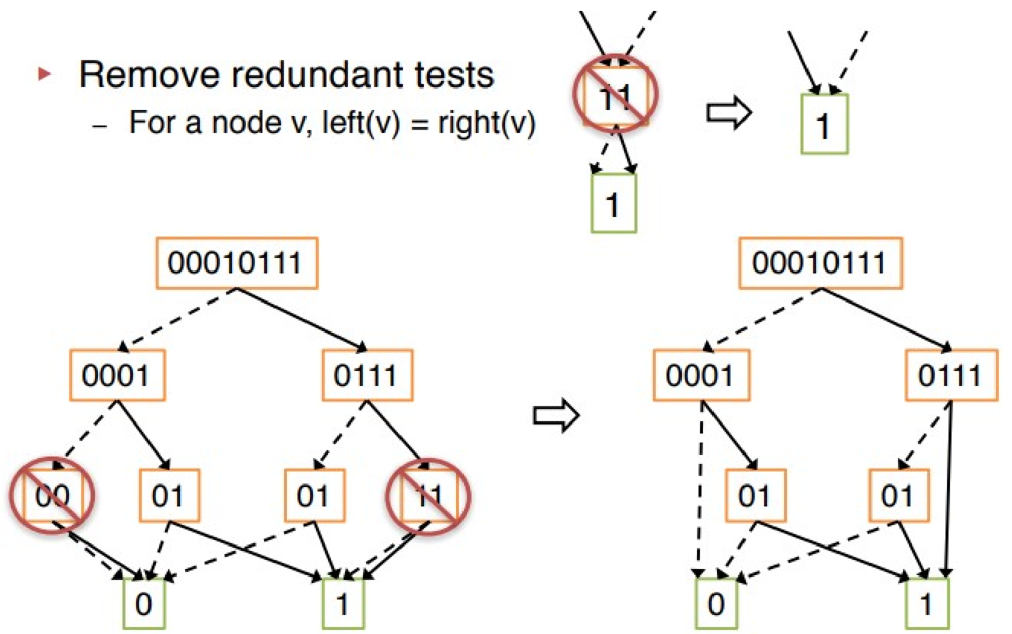
简化法则[8]：

- 合并等价叶子结点（只留0和1各一个）

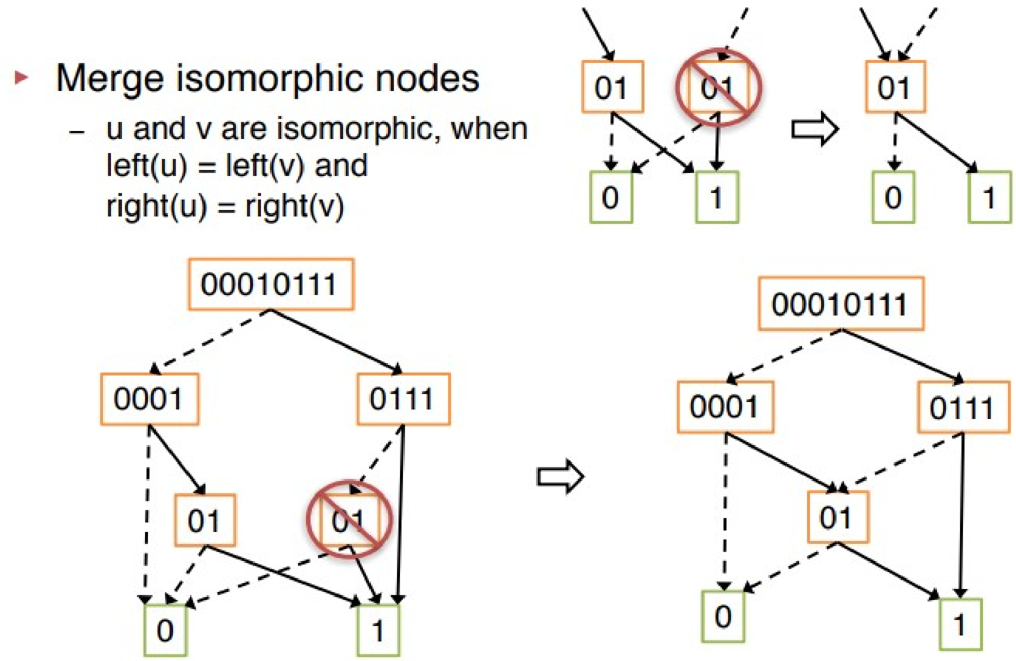
[8]图源来自Cornell ECE5775: https://www.csl.cornell.edu/courses/ece5775/pdf/lecture05.pdf

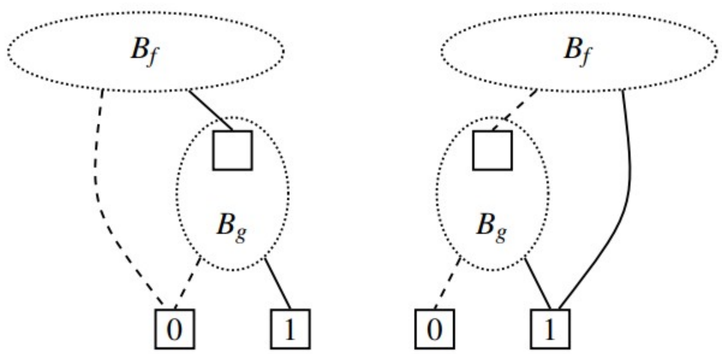


- 去除冗余测试(test)



- 合并同构结点



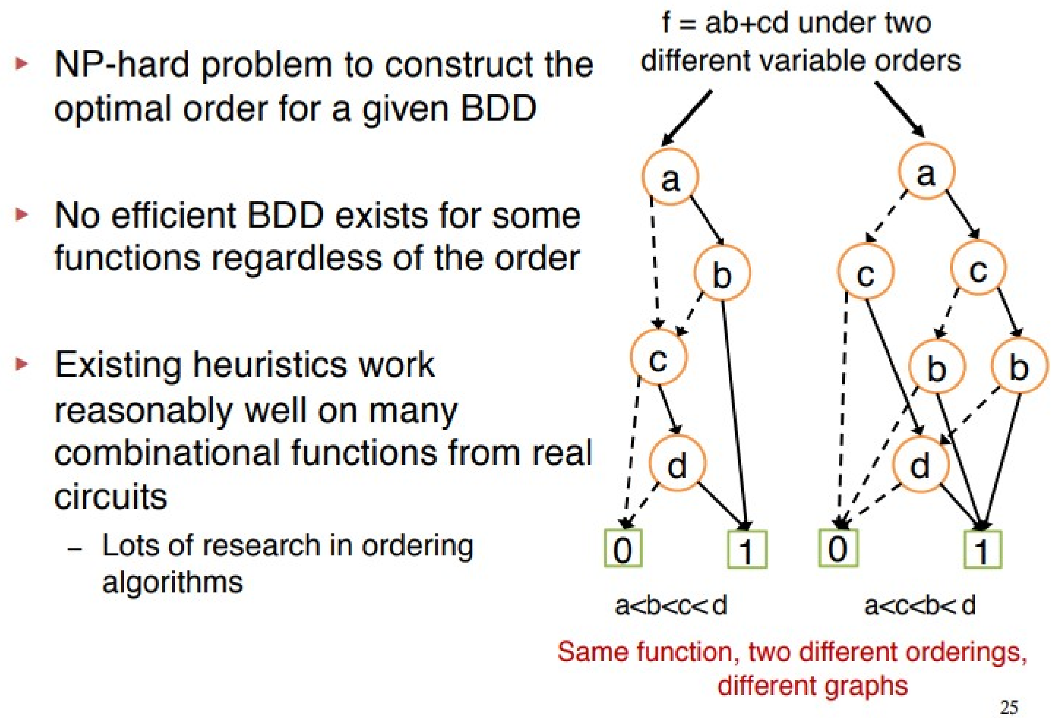


A BDD for $f \cdot g$ A BDD for $f + g$

定义 32 (有序BDD). 令$[x_1, \ldots, x_n]$为有序n个变量，B为含有这些变量的BDD

- 令$V(B)$为$OBDD$的变量集，即$V(B) = \{x_1, x_2, \ldots, x_n\}$
- $O_B(x_i)$是$\forall x_i \in V(B)$的序
- $\forall x_i, x_j \in V(B), O_B(x_i) < O_B(x_j)$，$B$中任一路径上$x_j$都在$x_i$后面

不同的序会带来不同的OBDD图，导致效率差异。



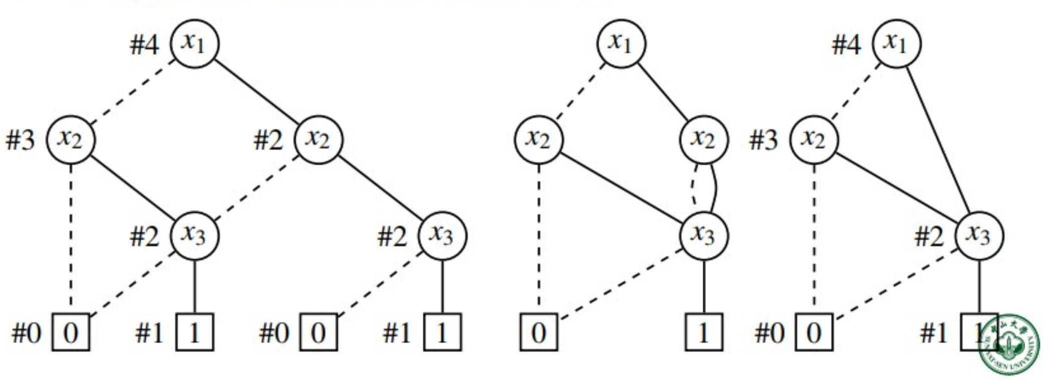
6.1 约简规则

定义 33. 给定BDD中非终端结点n

- $lo(n)$是从n用虚线指向的点
- $hi(n)$是从n用实线指向的点

$id(\cdot)$为结点编号

- If $id(lo(n)) = id(hi(n))$, we set $id(n)$ to be that label. That is because the boolean function represented at n is the same function as the one represented at $lo(n)$ and $hi(n)$.
- If $id(lo(n)) = id(lo(m))$ and $id(hi(n)) = id(hi(m))$, where n and m have the same variable x_i, we set $id(n)$ to be $id(m)$.
- Set $id(n)$ to the next unused integer label.



约简的BDD图和OBDD可在这个网页上在线画图。

6.2 应用规则

定义 34 (香农展开(Shannon expansion)). 对于所有布尔公式f及布尔变量x（包括那些不出现在f中的），有

$$f \equiv \bar{x} \cdot f[0/x] + x \cdot f[1/x]$$

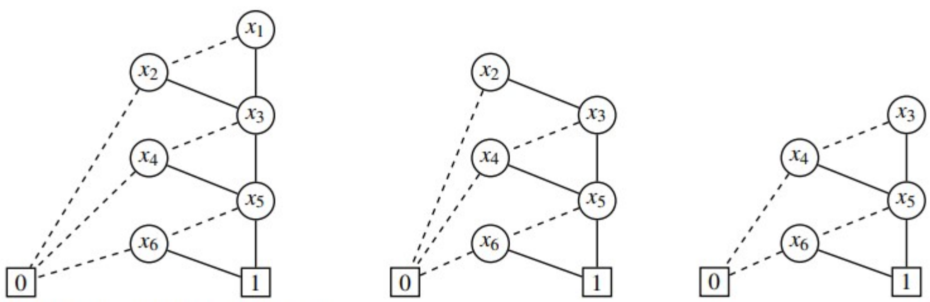
应用(apply)函数基于$f \;\; op \;\; g$的展开

$$\begin{aligned} f \;\; op \;\; g &\equiv (\bar{x} \cdot f[0/x] + x \cdot f[1/x]) \;\; op \;\; (\bar{x} \cdot g[0/x] + x \cdot g[1/x]) \\ &\equiv (\bar{x} \cdot f[0/x] \;\; op \;\; \bar{x} \cdot g[0/x]) + (x \cdot f[1/x] \;\; op \;\; x \cdot g[1/x]) \\ &\equiv \bar{x} \cdot (f[0/x] \;\; op \;\; g[0/x]) + x \cdot (f[1/x] \;\; op \;\; g[1/x]) \end{aligned}$$

6.3 总结

Let OBDD B_f represent formula f.

- **restrict**$(0, x, B_f)$ computes the reduced OBDD representing $f[0/x]$. For each node n labelled with x, incoming edges are redirected to $lo(n)$ and n is removed, and reduce on the resulting OBDD.
- **restrict**$(1, x, B_f)$ proceeds similarly, and redirect incoming edges to $hi(n)$.



The OBDD for $(x_1+x_2)\cdot(x_3+x_4)\cdot(x_5+x_6)$ with variable ordering $[x_1, x_2, x_3, x_4, x_5, x_6]$ $\quad$ $f[0/x_1] = x_2 \cdot (x_3 + x_4) \cdot (x_5 + x_6)$ $\quad$ $f[1/x_1] = (x_3 + x_4) \cdot (x_5 + x_6)$

- 特称规则：

$$\exists x.f \stackrel{\text{def}}{=\!=} f[0/x] + f[1/x]$$

- 全称规则：

$$\exists x.f \stackrel{\text{def}}{=\!=} f[0/x] \cdot f[1/x]$$

f	OBDD B_f	f	OBDD B_f
0	B_0	1	B_1
x	B_x	$\bar{f}$	交换B_f中的0和1结点
$f+g$	$apply(+, B_f, B_g)$	$f \cdot g$	$apply(\cdot, B_f, B_g)$
$f \oplus g$	$apply(\oplus, B_f, B_g)$		
$f[0/x]$	$restrict(0, x, B_f)$	$f[1/x]$	$restrict(1, x, B_f)$
$\exists x.f$	$apply(+, B_{f[0/x]}, B_{f[1/x]})$	$\forall x.f$	$apply(\cdot, B_{f[0/x]}, B_{f[1/x]})$