面向计算机科学的数理逻辑：系统建模与推理

Chapter 1 命题逻辑

逻辑符号

符号	含义	举例
¬	非	¬p
∨	析取（逻辑或）	p∨q
∧	合取（逻辑与）	p∧q
→	蕴含（推出）	p→q

证明规则

通过将**证明规则**(proof rules)应用于**前提**(premises)公式，**推断**(infer)出**结论**(conclusion)

ϕ1,ϕ2,...,ϕn⊢ψ 称为矢列(sequent)，如果可以找到矢列的证明，矢列称为有效的。若逻辑公式ϕ具有有效的矢列，称为**定理。**

一共15个演绎规则，其中11个基本规则，4个派生规则，**横线上是前提**，**横线下是结论**，**横线右边是记号**

合取规则：引入规则∧i，消去规则∧e_1, ∧e_2

$$\frac{\phi \quad \psi}{\phi \wedge \psi} \wedge i, \quad \frac{\phi \wedge \psi}{\phi} \wedge e_1, \quad \frac{\phi \wedge \psi}{\psi} \wedge e_2$$

双重否定规则： 引入规则(派生)¬¬i,消去规则¬¬e

$$\frac{\phi}{\neg\neg\phi} \neg\neg i, \quad \frac{\neg\neg\phi}{\phi} \neg\neg e$$

蕴含消去规则（分离规则）$\rightarrow e$

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

反证规则（派生）MT

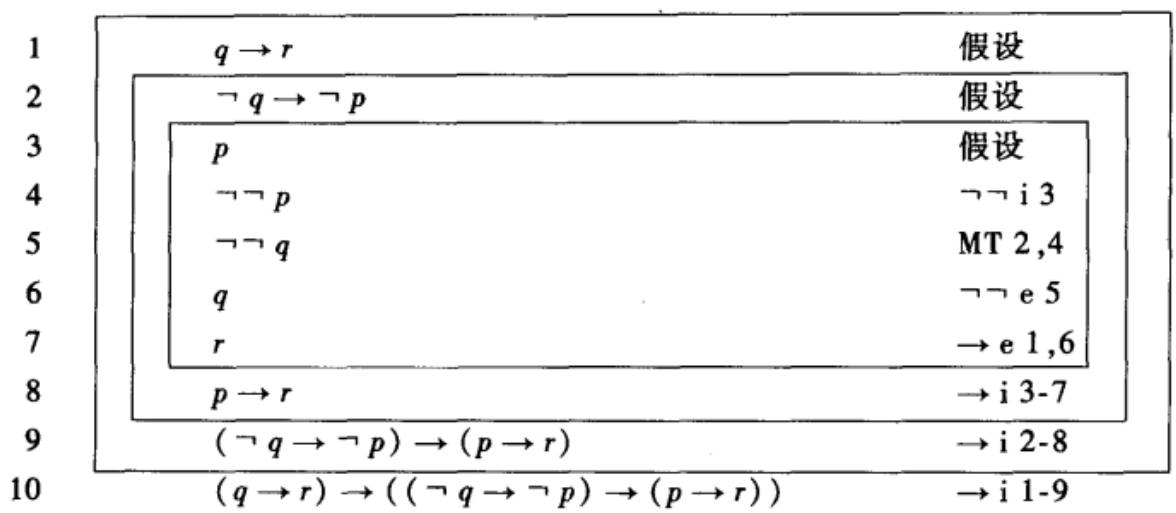
$$\frac{\phi \rightarrow \psi \quad \neg\psi}{\neg\phi}\mathrm{MT}$$

蕴含引入规则 $\rightarrow i$

$$\frac{\begin{bmatrix}\phi \\ \vdots \\ \psi\end{bmatrix}}{\psi} \rightarrow i$$

tips:这是为了证明$\phi \rightarrow \psi$，而临时假设了ϕ，然后再证明ψ，在证明ψ的过程中，可以用ϕ以及其他所有的公式，一般来说，只有在公式ϕ先于该位置出现，而且出现ϕ的矩形框都没有关闭的情况下，才可以使用ϕ。紧跟在关闭的矩形框后面的行必须与使用该矩形框的规则所得到的结论模式相匹配，即如果一个矩形框的第一个公式是ϕ，最后一个公式是ψ，那么紧跟在这个矩形框后面的行必须是$\phi \rightarrow \psi$

一个例子：



例 1.13 使用规则∧i，可以证明矢列 $p \wedge q \rightarrow r \vdash p \rightarrow (q \rightarrow r)$ 的有效性：

1	$p \wedge q \rightarrow r$	前提
2	p	假设
3	q	假设
4	$p \wedge q$	∧ i 2,3
5	r	→ e 1,4
6	$q \rightarrow r$	→ i 3-5
7	$p \rightarrow (q \rightarrow r)$	→ i 2-6

例 1.14 运用消去规则∧ e_1，∧ e_2，也可以证明上面矢列的“逆”的有效性：

1	$p \rightarrow (q \rightarrow r)$	前提
2	$p \wedge q$	假设
3	p	∧ e_1 2
4	q	∧ e_2 2
5	$q \rightarrow r$	→ e 1,3
6	r	→ e 5,4
7	$p \wedge q \rightarrow r$	→ i 2-6

$p \wedge q \rightarrow r \vdash p \rightarrow (q \rightarrow r)$ 和 $p \rightarrow (q \rightarrow r) \vdash p \wedge q \rightarrow r$ 的有效性说明这两个公式是等价的，即可以从一个证明另一个。把这种情况表示为：

$$p \wedge q \rightarrow r \dashv\vdash p \rightarrow (q \rightarrow r)$$

符号⊢的右边只对应一个公式，而符号⊣⊢对应两个公式。

析取规则 引入规则$\vee i_1, \vee i_2$，消去规则$\vee e$

$$\frac{\phi}{\phi \vee \psi} \vee i_1, \quad \frac{\psi}{\phi \vee \psi} \vee i_2, \quad \frac{\phi \vee \psi \quad \begin{bmatrix}\phi \\ \vdots \\ \chi\end{bmatrix} \quad \begin{bmatrix}\psi \\ \vdots \\ \chi\end{bmatrix}}{\chi} \vee e$$

矛盾规则$\rightarrow e$

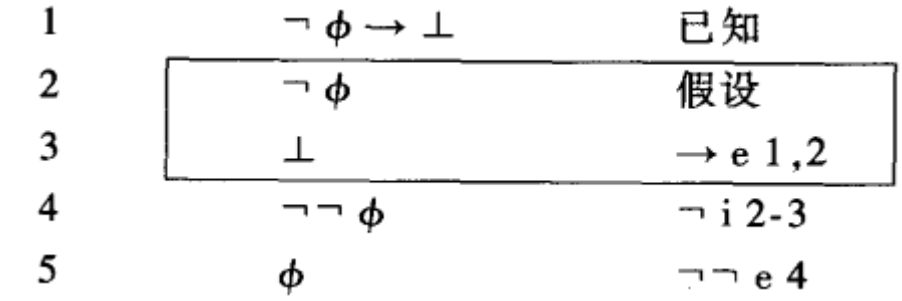
矛盾⊥是形如$\phi \wedge \neg\phi$或者$\neg\phi \wedge \phi$的表达式，矛盾可以推导出任何公式：

假设有前提p是 jf 有一个亿， q是jf会飞，那么p或q就肯定是真，但是非p 也就是jf没有一个亿为真，这时候p就是假，如果q也是假的话，那p或q就是假的，跟之前所说的有矛盾，因此q一定是真的，也就是说jf一定会飞

$$\frac{\bot}{\phi} \bot e \quad \frac{\phi \quad \neg\phi}{\bot} \neg e \quad \frac{\begin{bmatrix}\phi \\ \vdots \\ \bot\end{bmatrix}}{\neg\phi} \neg i$$

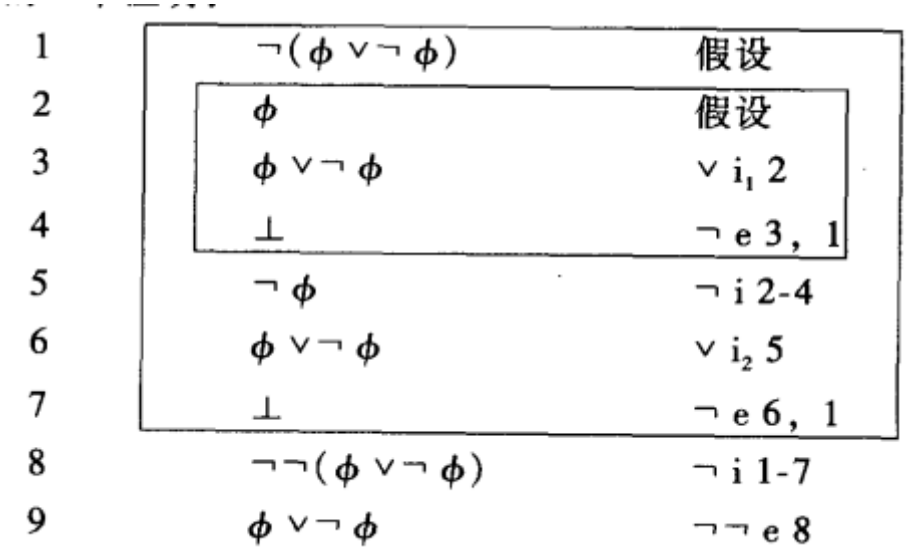
反证法（派生）PBC

$$\frac{\begin{bmatrix}\neg\phi \\ \vdots \\ \bot\end{bmatrix}}{\phi}\text{PBC}$$



排中律（派生）LEM

$$\frac{}{\phi \vee \neg\phi}\text{LEM}$$



以下六个式子逻辑等价：

$$\neg(p \wedge q) \dashv\vdash \neg q \vee \neg p \qquad \neg(p \vee q) \dashv\vdash \neg q \wedge \neg p$$
$$p \to q \dashv\vdash \neg q \to \neg p \qquad p \to q \dashv\vdash \neg p \vee q$$
$$p \wedge q \to p \dashv\vdash r \vee \neg r \qquad p \wedge q \to r \dashv\vdash p \to (q \to r)$$

作为形式语言的命题逻辑

合式公式（Backus Naur范式，BNF）其中p代表任意原子命题

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

可以用语法分析书表示和判定合成公式。最终叶子节点都是原子命题的树。

命题逻辑的语义

定义：公式ϕ的一个赋值(valuation)或模型(model)是对ϕ中的每个命题原子指派一个真值。两个公式的真值表相同，称为语义等价(semantically equivalent)。

数学归纳法：

1.用n=初始值开始证明

2.假设n=k时依然成立，当n=k+1时，带入并证明式子成立。

范式

语义等价：对与命题逻辑公式ϕ, ψ，说他们语义等价，当且仅当$\phi \models \psi$与$\psi \models \phi$成立。记为$\phi \equiv \psi$。进一步，如果$\models \phi$成立，称ϕ是有效的。也可以定义$\phi \equiv \psi$为$\models (\phi \to \psi) \wedge (\psi \to \phi)$。

由于合理性和完备性，语义等价和逻辑等价是一致的。

由于语义等价，我们希望把公式化成一种特定的形式，称作范式，在这种形式下有效性检查很容易。

Chapter 2 谓词逻辑

作为形式语言的谓词逻辑

谓词逻辑(predicate logic)也称一阶逻辑(first-order logic)，可以处理命题内部的逻辑结构，以及包含全称量词和存在量词的逻辑关系。
变量是实际值的一个占位符(place holder)

谓词词汇由两个集合组成：谓词符号集$\mathcal{P}$，函数符号集$\mathcal{F}$，常值看作零元函数nullary。
项(terms)：表示对象。定义：

·任何变量都是项。
·若$c \in \mathcal{F}$是零元函数，则c是项。
·若$t_1, t_2, \ldots, t_n$是项，且$f \in \mathcal{F}$的元$n > 0$，则$f(t_1, t_2, \ldots, t_n)$是项。
·没有其他类型的项。
BN范式：$t ::= x \mid c \mid f(t_1, t_2, \ldots, t_n)$，其中$x$取遍变量集合$var$，$c$取遍$\mathcal{F}$中的零元函数符号，$f$取遍$\mathcal{F}$中的$n > 0$的符号。

公式：表示真值。在$(\mathcal{F}, \mathcal{P})$定义：

·若$P \in \mathcal{P}$是$n \geq 1$元的谓词符号，$t_1, t_2, \ldots, t_n$是$\mathcal{F}$上的项，则$P(t_1, t_2, \ldots, t_n)$是公式。
·若ϕ, ψ是公式，则$(\neg\phi), (\phi \wedge \psi), (\phi \vee \psi), (\phi \to \psi)$也是公式。
·若ϕ是公式，x是变量，则$(\forall x\phi), (\exists x\phi)$也是公式。
·没有其他形式的公式。

谓词逻辑公式可以用语法分析树表示。例如，图 2-1 的分析树表示公式 $\forall x((P(x) \to Q(x)) \wedge S(x, y))$。

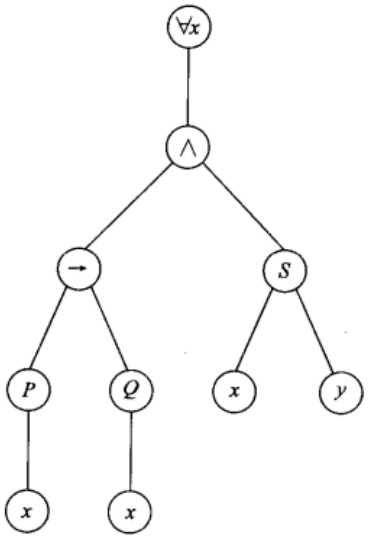


图 2-1　一个谓词逻辑公式的语法分析树

自由变量和约束变量 设ϕ是谓词逻辑中的公式。称x在ϕ中的一次出现是自由的，如果x是ϕ的语法分析树中的一个叶结点，而且不存在从结点x到$\forall x$或$\exists x$的向上路径。否则，称x的出现是约束的。对$\forall x$或$\exists x$，我们称除去ϕ的任何形如$\forall x$或$\exists x$的子公式的ϕ分别是$\forall x$或$\exists x$的作用范围。

变量代换 给定变量x、项t、和公式ϕ，定义$\phi[t/x]$为用t代替ϕ中变量x的每个自由出现而得到的公式。
给定变量x、项t、和公式ϕ，说t关于ϕ中的x是自由的，如果对出现在t中的任何变量y，ϕ中没有自由的叶结点x处于$\forall y$或$\exists y$的作用范围之内。

谓词逻辑的证明论

自然演绎规则

约定：$\phi[t/x]$总假定t对ϕ中x是自由的。
在命题逻辑自然演绎基础上加上：

相等的证明规则：

$$\frac{}{t=t}=i, \qquad \frac{t_1=t_2 \quad \phi[t_1/x]}{\phi[t_2/x]}=e$$

全称量词的证明规则：

$$\frac{\begin{bmatrix} x_0 \\ \vdots \\ \phi[x_0/x] \end{bmatrix}}{\forall x\phi}\forall xi, \qquad \frac{\forall x\phi}{\phi[t/x]}\forall xe$$

存在量词的证明规则：

$$\frac{\phi[t/x]}{\exists x\phi}\exists xi, \qquad \frac{\exists x\phi \quad \begin{bmatrix} x_0\phi[x_0/x] \\ \vdots \\ \chi \end{bmatrix}}{\chi}\exists xe$$

量词等价(假设x在ψ中是不自由的)：

$$\begin{array}{rl}
\neg\forall x\phi \dashv\vdash \exists x\neg\phi & \neg\exists x\phi \dashv\vdash \forall x\neg\phi \\
\forall x\phi \wedge \psi \dashv\vdash \forall x(\phi\wedge\psi) & \forall x\phi \vee \psi \dashv\vdash \forall x(\phi\vee\psi) \\
\exists x\phi \wedge \psi \dashv\vdash \exists x(\phi\wedge\psi) & \exists x\phi \vee \psi \dashv\vdash \exists x(\phi\vee\psi) \\
\forall x(\psi\to\phi) \dashv\vdash \psi\to\forall x\phi & \exists x(\psi\to\phi) \dashv\vdash \psi\to\exists x\phi \\
\exists x(\phi\to\psi) \dashv\vdash \forall x\phi\to\psi & \forall x(\phi\to\psi) \dashv\vdash \exists x\phi\to\psi \\
\forall x\phi\wedge\forall x\psi \dashv\vdash \forall x(\phi\wedge\psi) & \exists x\phi\vee\forall x\psi \dashv\vdash \exists x(\phi\vee\psi) \\
\forall x\forall y\phi \dashv\vdash \forall y\forall x\phi & \exists x\exists y\phi \dashv\vdash \exists y\exists x\phi
\end{array}$$

谓词逻辑的语义

模型 假设$\mathcal{F}$是函数符号的集合，$\mathcal{P}$是谓词符号的集合，每个符号所需要的变量个数是固定的。符号对$(\mathcal{F},\mathcal{P})$的一个模型$\mathcal{M}$由下面的数据集合组成：

1.非空集A是具体值的全集;

2.对每个零元函数$f\in\mathcal{F}$，A 中的一个具体元素$f^{\mathcal{M}}$

3.对每个元数为$n>0$的$f\in\mathcal{F}$，从集合A上n元集合A^n 到A的具体函数$f^{\mathcal{M}}:A^n\to A$

4.对每个$n>0$元谓词$P\in\mathcal{P}$，A上n元子集$P^{\mathcal{M}}\subseteq A^n$ 。

环境 从变量集var到相关模型中值的论域集合A的函数$l:\mathbf{var}\to A$。对这样的l，用$l[x\mapsto a]$表示将x映到a并且将其他变量y映到$l(y)$的查询表。

给定关于对$(\mathcal{F},\mathcal{P})$的模型$\mathcal{M}$和环境$l$，对于$(\mathcal{F},\mathcal{P})$上每个逻辑公式$\phi$，通过对$\phi$的结构归纳定义一个满足关系$\mathcal{M}\models_l\phi$。若$\mathcal{M}\models_l\phi$成立，则称在模型$\mathcal{M}$中，相对于环境$l$，$\phi$的赋值为T。

P：如果ϕ的形式为$P(t_1,t_2,\ldots,t_n)$，则在集合A中将$t_1,t_2,\ldots,t_n$解释为：把所有变量根据l的值代替。用这种方式，对每项（通过$f\in\mathcal{F}$）计算$a_1,a_2,\ldots,a_n$的值，其中任何函数符号$f\in\mathcal{F}$通过$f^{\mathcal{M}}$ 来0+解释。现在$\mathcal{M}\models_l P(t_1,t_2,\ldots,t_n)$成立当且仅当$(a_1,a_2,\ldots,a_n)\in P^{\mathcal{M}}$
$\forall x$：关系$\mathcal{M}\models_l\forall x\psi$成立当且仅当$\mathcal{M}\models_{l[x\mapsto a]}\psi$对所有$a\in A$都成立。
$\exists x$：对偶地，$\mathcal{M}\models_l\exists x\psi$成立当且仅当$\mathcal{M}\models_{l[x\mapsto a]}\psi$对某个$a\in A$成立。
$\neg$：关系$\mathcal{M}\models_l\neg\psi$成立当且仅当$\mathcal{M}\models_l\psi$不成立。
$\vee$：关系$\mathcal{M}\models_l\psi_1\vee\psi_2$成立当且仅当$\mathcal{M}\models_l\psi_1$成立或$\mathcal{M}\models_l\psi_2$成立。
$\wedge$：关系$\mathcal{M}\models_l\psi_1\wedge\psi_2$成立当且仅当$\mathcal{M}\models_l\psi_1$和$\mathcal{M}\models_l\psi_2$都成立。
$\to$：关系$\mathcal{M}\models_l\psi_1\to\psi_2$成立当且仅当只要$\mathcal{M}\models_l\psi_1$成立，则$\mathcal{M}\models_l\psi_2$成立。

语义推导

Γ是谓词逻辑中的公式集合（可能是无限集合），ψ是谓词逻辑公式。

语义推导$\Gamma\models\psi$成立当且仅当对所有的模型$\mathcal{M}$和查询表l，对所有的$\phi\in\Gamma$，$\mathcal{M}\models_l\phi$都成立，则$\mathcal{M}\models_l\psi$也成立。
公式ψ是可满足的当且仅当存在某个模型$\mathcal{M}$和环境l，使得$\mathcal{M}\models_l\psi$成立。
公式ψ是有效的当且仅当在我们能够检测ψ的所有模型$\mathcal{M}$和环境l中，$\mathcal{M}\models_l\psi$成立。
集合Γ是一致的或可满足的当且仅当存在一个模型$\mathcal{M}$和一个环境l，使得对所有的公式$\phi\in\Gamma$，$\mathcal{M}\models_l\psi$成立。

谓词逻辑的不可判定性

方法：问题归约法。选择另一个不可解问题，用当前问题的可解性推出被选择问题的可解性。
引理：波斯特对应问题不可解。波斯特对应问题：已知有限序列对$(s_1,t_1),(s_2,t_2),\ldots,(s_k,t_k)$，其中所有的$s_i,t_i$ 都是正长度的二进制字符串。是否存在下标序列$i_1,i_2,\ldots,i_n(n\geq1)$，使得字符串的串联$s_{i1}s_{i2}\ldots s_{im}$等于$t_{i1}t_{i2}\ldots t_{im}$?

原定理证明：

假设谓词逻辑有效性可判定。求解波斯特对应问题。已知对应问题的实例C：$s_1s_2\ldots s_k,t_1t_2\ldots t_k$。需要在有限的时空里构建对所有实例都一致的谓词逻辑公式ϕ，使得$\models\phi$成立，当且仅当波斯特对应问题C有一个解。
构造公式(这里是整个证明的核心，公式不好理解的话在必要性那里有解释)：
选择一个常量e，以及两个需要一个自变量的函数符号f_0和f_1。把e视为空位符串或空字，f_0和f_1分别代表与0和1拼接。若$b_1b_2\ldots b_l$是二进制位的字符串，将它编码为项$f_{b_l}(f_{b_{l-1}}\ldots(f_{b_2}(f_{b_1}(e)))\ldots)$，简记为$f_{b_1b_2\ldots b_l}(e)$
谓词符号$P(v,w)$表示存在某个下标序列$(i_1,i_2,\ldots,i_m)$，使得v是由$s_{i_1}s_{i_2}\ldots s_{i_m}$表示的项，$w$是由$t_{i_1}t_{i_2}\ldots t_{i_m}$表示的项。$v$和$w$使用同样的下标序列构造出一个串；只是$v$使用$s_i$，$w$使用$t_i$。
语句ϕ有结构：$\phi_1\wedge\phi_2\to\phi_3$，其中，设

$$\phi_1\stackrel{\text{def}}{=}\bigwedge_{i=1}^{k}P(f_{s_i}(e),f_{t_i}(e))$$

$$\phi_2\stackrel{\text{def}}{=}\forall v\forall w\left(P(v,w)\to\bigwedge_{i=1}^{k}P(f_{s_i}(v),f_{t_i}(w))\right)$$

$$\phi_3\stackrel{\text{def}}{=}\exists zP(z,z)$$

论断$\models\phi$成立，当且仅当波斯特对应问题C有解。

必要性：

太烦不证了。

谓词逻辑的表达能力

软件模型、设计标准和硬件或程序的执行模型经常通过有向图来描述。这样的模型$\mathcal{M}$就是定义在具体"状态"集合A上的二元谓词R的解释。

例：给定状态集合$A=\{s_0,s_1,s_2,s_3\}$，令$R^{\mathcal{M}}$是集合
$\{(s_0,s_1),(s_1,s_0),(s_1,s_1),(s_1,s_2),(s_2,s_0),(s_3,s_0),(s_3,s_2)\}$。用有向图描述这个模型，当且仅当$(s,s')\in R^{\mathcal{M}}$，存在一条从结点$s$，到结点$s'$的边。在这种情形下，通常记作$s\to s'$。

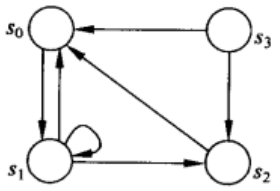


图 2-5　一个有向图，是两个参量的谓词符号 R 的模型 $\mathcal{M}$。一个结点对 (n, n') 在 R 的解释 $R^{\mathcal{M}}$ 中，当且仅当图中有一个从结点 n 到结点 n' 的迁移（边）

紧致性定理 设Γ是谓词逻辑的一个语句集合。若Γ的所有有限子集都是可满足的，则Γ也是可满足的。

Löwenheim-Skolem 定理 设ψ是谓词逻辑的一个语句，对任何自然数 $n\geq 1$，存在ψ的至少有n个元素的模型。则ψ有无限个元素的模型。

可达性：给定有向图中的结点n和n'，是否存在从n到n'的有限长度的迁移路径?

可达性用谓词逻辑是不可表达的：不存在仅以u和v为自由变量且仅有一个二元谓词符号R的谓词逻辑公式ϕ，使得ϕ在有向图中成立当且仅当该有向图中存在一条从伴随u的结点到伴随v的结点的路径。

使用二阶逻辑表达可达性：将量词应用于谓词。

Chapter 3 通过模型检测进行验证

模型检测

时态逻辑（Temporal Logic）
思想：在一个模型中，公式的真与假不是静态的，而在命题逻辑或谓词逻辑中的确如此。
含义：时态逻辑的模型包含若干状态，而一个公式可以在某些状态下为真，在其他状态下为假。公式可以随系统的状态演化而改变其真值。
模型检测：模型检测是一种自动的、基于模型的、性质验证的处理方法。
模型检测也是基于时态逻辑的，在模型检测中，模型$\mathcal{M}$是迁移系统，性质ϕ是时态逻辑公式，为了验证一个系统满足一个性质，需要三步操作：
使用模型检测器的描述性语言对系统进行建模，得到一个模型$\mathcal{M}$
使用模型检测器的规范语言对性质进行编码，禅师一个时态逻辑公式ϕ
以$\mathcal{M}$和ϕ做输入，运行模型检测器

模型检测就是对问题$\mathcal{M},s\models\phi$是否成立计算答案的过程，此处的ϕ是时态逻辑的一个公式，$\mathcal{M}$是所考虑的一个适当模型，s是该模型的一个状态，$\models$是满足关系

线性时态逻辑

定义：一种能表达时间概念的特殊时态逻辑。它将时间轴看成一个线性的状态序列，可以无限延伸到未来。常用来精确表示模型的动态语义。
计算路径（简称为路径）：模型里的状态序列。因为未来是不确定的，有无数种可能，在模型中表示有无数条路径，代表未来不同的可能，任何一条路径都可能会是一条实际的路径（每一条路径都有可能发生，但只有一条路径会真正发生）。
命题原子公式：原子公式用$p, q, r, p_1, q_1, r_1 \ldots$等符号表示，这些原子代表系统可能成立的事实，比如“打印机正在打印”，“进程1220被挂起”，或者“程序计数器上的值为6”。
原子集合：表示系统可能成立的全部事实，用$Atoms$表示，比如p, q的所有组合：
$Atoms = \{\varnothing, \{p\}, \{q\}, \{p, q\}\}$

LTL语法

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

上述公式表示，如果ϕ是LTL公式，则$\neg\phi, (\phi \wedge \phi), (G\phi) \cdots$也是LTL公式，同时$\bot \mid \top \mid p$也是LTL公式。其中，$p$是取自原子集合$Atoms$的任意命题原子

时态连接词：

一元时态连接词：
X：下一个状态。
F：某未来状态。
G：所有未来状态。
二元时态连接词：
U：直到。
R：释放。
W：弱-直到。

时态连接词的含义：

原子命题p表示路径π中第一个原子命题是p

Xp表示路径π中第二个原子命题是p

Gp表示路径π中每一个原子命题都是p

pUq表示对于路径π：$s_0 \rightarrow s_1 \rightarrow \cdots$中，假设原子命题$p$在且只在$s_3, s_4, s_5, s_6, s_7, s_8$状态点满足，原子命题$q$只在$s_9$状态点满足。则如果$i$的值为$0, 1, 2$，$\pi i \nvDash pUq \pi^i \nvDash pUq$（因为这个时候的开始点没有$p$）；如果$i$的值为$3, 4, 5, 6, 7, 8, 9$，则$\pi^i \models pUq$。

pRq:q必须保持为真，直到p为真的时刻

或者
如果p为真的时刻不存在，则q一直为真

R是U的对偶，即$\phi R \psi$等价于$\neg(\neg\phi U \neg\psi)$

LTL公式范例：$(F(p \rightarrow (G\ r)) \vee ((\neg q) U\ p))$