Principle & Applications of Artificial Intelligence

Chapter 1
Rule-Based Reasoning
规则推理
(Logical Reasoning)

Contents

1.1 First-Order Logic System
一阶谓词逻辑系统

- 1.1.1 知识与逻辑语句
- 1.1.2 一阶谓词逻辑系统
- 1.1.3 逻辑系统的语法和语义

1.1.1 知识与逻辑语句

- 知识(knowledge)用句子(sentence)来表示
- 选用陈述句(statement)表示事物真假(真理/知识)
  - 哈尔滨今天天气晴　　（晴朗则T；阴天则F）
  - 哈尔滨的天多么晴朗啊！　　（非陈述句）
- 使用具有真假值(分别用T/F表示)判断一个陈述句或者一个词表示一个命题（用P、Q等符号）
- 一个命题是一个原子命题；原子命题用操作符联结起来构成复合命题；用复合命题表示的内容和操作称为命题逻辑
- 命题中将所有的陈述视为一个整体，不区分内部结构

谓词公式

- 将命题中用于陈述或描述的谓词与被描述/操作的对象分开表示，构成了谓词语句或称为谓词公式
  - 天气晴(哈尔滨，今天)　　T
  - 上课(周三5-6节，诚意11，AI)　　T
  - 上课(周三1-2节，诚意11，AI)　　F
- 一个单个的谓词称为原子公式；原子公式通过连接符等组成复合公式；复合公式的演算/推导称为谓词逻辑，表示如下：
  - C(t, r, n)　　其中C表示”上课”，t/r/n分别表示时间、地点(课程教室)、课程名，均为变量

谓词的论域

- 谓词可看作是从个体域（定义域）或个体域的笛卡儿乘积到真值集合{T/F}的映射，个体域通常称为论域
- 例子：
  - 人都会死亡，可以写为 M(x)→D(x)，其论域是人；给定了论域，就确定了谓词真假值
  - 诚意楼教室在一天中总会有空闲的时刻，其论域是两个论域{r}{t}的复合—笛卡尔乘积，表示为：
  - Unoccupied(r, t) 当某个教室在某个学时中为空闲时，该谓词为真
- 一元谓词表示性质，二元或多元谓词表示关系

谓词公式举例

- 有人能够在某一时间欺骗(deceive)所有人，也能够在所有时间欺骗某个人；但是没有人能够在所有时间欺骗所有人。
  - $\exists x\ \exists t\ \forall y\ M(x) \wedge M(y) \wedge T(t) \wedge D(x, y)$
  - $\exists x\ \exists y\ \forall t\ M(x) \wedge M(y) \wedge T(t) \wedge D(x, y)$
  - $\neg \exists x\ \forall t\ \forall y\ M(x) \wedge M(y) \wedge T(t) \wedge D(x, y)$
- 所有喜爱动物的人都会被某个人所喜爱
  - $\exists x\ (\forall y\ M(y) \wedge Love(y, z) \wedge Animal(z) \rightarrow Love(x, y))$
- 质数(prime)可以被1和自身整除(余数remainder为0，除法division)（如果换为：只能被…）
  - $P(x) \rightarrow D(x, 1, 0) \wedge D(x, x, 0)$ 表示余数=0的除法

1.1.2 一阶谓词逻辑系统

- 一阶谓词逻辑符号体系

原子公式

复合公式

$$
\begin{aligned}
\textit{Sentence} &\rightarrow \textit{AtomicSentence} \mid \textit{ComplexSentence} \\
\textit{AtomicSentence} &\rightarrow \textit{Predicate} \mid \textit{Predicate}(\textit{Term}, \ldots) \mid \textit{Term} = \textit{Term} \\
\textit{ComplexSentence} &\rightarrow (\textit{Sentence}) \mid [\textit{Sentence}] \\
&\mid \neg \textit{Sentence} \\
&\mid \textit{Sentence} \wedge \textit{Sentence} \\
&\mid \textit{Sentence} \vee \textit{Sentence} \\
&\mid \textit{Sentence} \Rightarrow \textit{Sentence} \\
&\mid \textit{Sentence} \Leftrightarrow \textit{Sentence} \\
&\mid \textit{Quantifier}\ \textit{Variable}, \ldots\ \textit{Sentence}
\end{aligned}
$$

项

$$
\begin{aligned}
\textit{Term} &\rightarrow \textit{Function}(\textit{Term}, \ldots) \\
&\mid \textit{Constant} \\
&\mid \textit{Variable}
\end{aligned}
$$

量词

常量

变量

谓词

函数

操作优先级

$$
\begin{aligned}
\textit{Quantifier} &\rightarrow \forall \mid \exists \\
\textit{Constant} &\rightarrow A \mid X_1 \mid \textit{John} \mid \cdots \\
\textit{Variable} &\rightarrow a \mid x \mid s \mid \cdots \\
\textit{Predicate} &\rightarrow \textit{True} \mid \textit{False} \mid \textit{After} \mid \textit{Loves} \mid \textit{Raining} \mid \cdots \\
\textit{Function} &\rightarrow \textit{Mother} \mid \textit{LeftLeg} \mid \cdots \\
\text{OPERATOR PRECEDENCE} &: \neg, =, \wedge, \vee, \Rightarrow, \Leftrightarrow
\end{aligned}
$$

为什么称为一阶谓词逻辑？

- 命题中不区分谓词和个体，可以视作0阶逻辑
- 一阶/1阶谓词逻辑区分了谓词和个体（对象），并且引入了作用于个体的量词(∃/∀)
  - 对于诚意楼的教室，在一天的某个学时总有一间是空闲的
  - $\exists r\, \exists t$ Classroom(r, 诚意楼)$\wedge$Class-hour(t)$\wedge$ Unoccupied(r, t)
- 2阶谓词逻辑对谓词也进行量化，如个体集合的任意性质P（作用于个体集合之上），例子如下：
  - $\forall x, y\ (x=y \rightarrow \forall P\ P(x) \Leftrightarrow P(y))$

量词的使用

- 约束变量是∀、∃中x的变量，量词所管辖的公式如P(x)称为量词辖域；不在量词辖域内的变量为自由变量
- 区别：自由变量可代入常量，约束变量不行，因为∀a P(a)无意义；约束变量可改名，自由变量不行
- 带有全称变量∀x的公式，表示形式为∀x(P(x)→…)
- 带有存在变量∃x的公式，表示形式为∃x(P(x)∧…)

一阶语言

- 一阶语言是一阶逻辑使用的形式语言，记为L；可以和任何论域没有联系，也可以与某个论域有联系

■ 与论域没有联系的一阶语言由8类符号组成：

(1)无限序列的个体符号（个体常量）； (2)无限序列的谓词符号，有确定的元数n⩾1；一个特殊的谓词符号称为相等符号（等式），记为=；L中可含或不含=，如果含有，即称为含=的一阶逻辑；(3)无限序列的函数符号，有确定的元数m≥1；

(4)无限序列的自由变量：可用u/v/w等表示自由变量；(5)无限序列的约束变量：可用x/y/z等表示约束变量；

(6)联结词：¬∧∨→≡；(7)量词：∀、∃；

(8)标点：左右括号、逗号

一阶语言的项和原子公式

- *L*的项：一阶语言中的一个符号是项t，当且仅当它能通过有限次使用下述步骤生成：

  (1)个体常量、自由变量是项；

  (2)如果$t_1 \ldots t_n$是项，且f是n元函数，则$f(t_1 \ldots t_n)$是项

- *L*的原子公式：一阶语言中的一个表达式是一个原子公式，当且仅当它有如下2种形式：

  (1)$F(t_1 \ldots t_n)$，F是n元谓词，$t_1 \ldots t_n$是项；

  (2)$=(t_1,t_2)$或$t_1=t_2$，t_1、t_2是项

一阶语言的公式

- *L*的公式：一阶语言中的一个表达式是一个公式，当且仅当它能通过有限次使用下述步骤生成：

  (1)原子公式是公式；

  (2)如果A是公式，则(¬A)是公式；

  (3)如果A、B均为公式，则A*B是公式，

  其中*表示∧∨→≡中的任意一个；

  (4)如果A(u)是公式，且x不在A(u)中出现，则∀x A(x)、∃x A(x)都是公式

形式系统

- 一阶谓词逻辑系统(也叫形式系统Formal System，FS)由5个部分组成：

  (1)符号表—非空集合Σ，即一阶逻辑语言；

  (2)Σ上全体符号的集合Σ*的子集—项和变量；

  (3)Σ*的子集—公式，公式和项的交集为空；

  (4)公式FORMULA上的子集—公理AXIOM；

  (5)公式上的n元关系集合—推理规则RULE

- 符号表Σ、项TERM、FORMULA称为FS的组成部分；AXIOM、RULE称为FS的推演部分；逻辑系统中除了符号表以外的部分构成了逻辑系统的语法

1.1.3 逻辑系统的语法和语义

- 形式推演定义了公式之间的形式可推演性关系，它涉及公式语法结构，其正确性能够机械地证明：用记号$\vdash$表示形式可推演关系，读作“推出”；用$\Sigma \vdash A$表示A是由Σ形式可推演的（或形式可证明的），其中Σ是前提，A是结论
- 形式可推演性：在一阶谓词逻辑系统中，A是由Σ可形式推演的，记为$\Sigma \vdash A$，当且仅当$\Sigma \vdash A$能通过有限次应用相应逻辑的形式推演规则生成，即：$\Sigma \vdash A$成立，当且仅当存在有限序列使得$\Sigma_1 \vdash A_1$，$\Sigma_2 \vdash A_2$，…，$\Sigma_n \vdash A_n$中的每一项均由某个形式推导规则生成，且$\Sigma_n \vdash A_n$，即$\Sigma_1=\Sigma$，$A_n=A$

逻辑系统的语义

- 逻辑系统的语义是对可推演性进行解释，研究公式的可满足性和有效性；研究逻辑系统的语义时，对逻辑系统赋予研究对象的集合（即论域）及上面的运算；此时每个公式都有具体含义（语义），因此其真假赋值是有具体约束的
- 研究语义时，对逻辑系统赋予论域；用论域中的个体对象、对象之上的运算(函数)、对象之间的关系(谓词)去解释逻辑系统中的符号
- 解释就是判断谓词公式的真假值

可满足性

- 可满足性：Σ是可满足的，当且仅当有真假赋值v，使得$\Sigma^v=1$；此时称v满足Σ
- Σ的可满足性蕴涵了Σ中所有公式的可满足性，但反过来不成立；因为这要求同一个真假赋值满足所有的公式，但是实际上并非所有可满足的公式都使用同一个赋值
- 逻辑推论：设Σ、A分别是一阶语言的公式集和公式，A是Σ的逻辑推论，记作$\Sigma|=A$，当且仅当对于任何赋值v，$\Sigma^v=1$蕴涵$A^v=1$
- 在一阶谓词逻辑系统中，形式推演是语法概念，逻辑推论是语义概念

公式集合的可满足性

- 给定公式某个赋值(为真)是可满足性；给定任意赋值(为真)则是有效性
- 公式集合的可满足性：一阶逻辑公式集合Σ是可满足的，当且仅当有(以某个非空集为论域)赋值v，使得$\Sigma^v=1$（为真）；当$\Sigma^v=1$时，称v满足Σ
- 反过来，不可满足性就是对任意论域上的任意赋值都有$\Sigma^v=0$

有效性

- 有效性：一阶逻辑公式A是有效的（或者称为合法的），当且仅当对于(以任何不空集为论域)任何赋值v，$A^v = 1$
- 论域中的可满足性、有效性：

(1)Σ在D中是可满足的，当且仅当对于以D为论域的赋值v，使得$\Sigma^v = 1$

(2)A在D中是有效的，当且仅当对于任何以D为论域的赋值v，$A^v = 1$

语法和语义之间关系：整体特征

- 逻辑系统的整体特征：逻辑系统在引入语义以后，需要研究形式推演（语法）与逻辑推论（语义）之间的关系；即研究：对于任何一阶语言的公式集Σ和公式A在何种赋值的条件下，其结果为真
- 不仅要研究逻辑系统的推导过程，而且要研究在特定论域中的可满足性（一般是公式集）、有效性（一般是公式）

语法推导与语义解释

- 对于任何一阶语言的公式集Σ和公式A，有以下关系：
- $\Sigma \vdash A \Rightarrow \Sigma \models A$表示：凡是形式可推演性所反映的前提和结论之间的关系，在非形式演算(如赋值)中都是成立的，即形式可推演性对于反映非形式演算/赋值是可以保证正确的，此即可靠性定理(或称合理性定理)
- $\Sigma \models A \Rightarrow \Sigma \vdash A$表示：凡是在非形式演算（论域赋值判断）中成立的前提和结论之间的关系，形式可推演性都是能够反映的，即形式可推演性在反映非形式演算时没有遗漏，此即完备性定理

可靠性定理和完备性定理

- 可靠性定理：设$\Sigma \subseteq Form(L)$（Σ为一阶语言的公式集合），$A \in Form(L)$ (也包括了命题语言)

  (1)如果$\Sigma \vdash A$，则$\Sigma \models A$；

  (2)如果$\varnothing \vdash A$，则$\varnothing \models A$(即所有形式可推演的都是有效的)

- 完备性定理：设$\Sigma \subseteq Form(L)$，$A \in Form(L)$(也包括命题语言)

  (1)如果$\Sigma \models A$，则$\Sigma \vdash A$；

  (2)如果$\varnothing \models A$，则$\varnothing \vdash A$(所有有效公式都是形式可证明公式)

1.2 Inference in FOL
一阶谓词逻辑的推理

- 1.2.1 谓词逻辑推理的基本操作
- 1.2.2 前向链接和后向链接算法
- 1.2.3 谓词公式向子句转化
- 1.2.4 消解算法

1.2.1 谓词逻辑推理的基本操作

- 推理规则的一般形式
- $p_1 \wedge \ldots \wedge p_k \rightarrow q$ 前提→结论
- **具体应用：规则条件为$p_i(x)$形式，**如果有条件$p_i(b)$存在，那么是否满足规则条件呢？
- 形式上需要使$p_i(x)$与$p_i(b)$一致，于是引入置换与合一操作：
- 对于谓词中的个体，以项来代替变量，就是置换；而使两个同一谓词的个体置换为相同个体，就是合一置换；完成了合一置换，两个谓词完全相同，即满足了前提，可以继续推理

置换与合一置换

- 置换（代换）（substitution）：设$x_1 \ldots x_n$是n个变量，且各不相同，$t_1 \ldots t_n$是n个项(常量、变量、函数)，$t_i \neq x_i$，则用t_i替换变量x_i操作形成的有限序列$\{x_1/t_1, \ldots, x_n/t_n\}$称为一个置换（运算）
  - 置换合成：设θ和λ是2个置换，则先θ后λ作用于公式或项，称为置换合成(乘积)，用θ°λ表示
  - 通过相关置换使不同的一阶谓词公式成为相同的过程称为合一（合一置换）
- 合一置换（unification）：设有一组谓词公式$\{F_1, \ldots, F_k\}$和置换θ，使得$F_1\theta = F_2\theta = \cdots = F_k\theta$，则θ称为合一置换(具体形式就是一个置换序列)，$F_1, \ldots, F_k$称为可合一的

最一般合一置换

- 最一般合一置换（mgu，most general unification）：如果σ和θ都是公式组$\{F_1, \ldots, F_k\}$的合一置换，且有置换λ存在，使得θ=σ°λ，则σ称为公式组$\{F_1, \ldots, F_k\}$的最一般合一置换（实际上就是最简单的），记为mgu
- 合一结果是一个置换，如UNIFY(Know(John, x), Know(John, Jane)) ={x/Jane}，UNIFY(Know(John, x), Know(y, Mary)) = {x/Mary, y/John}；但是UNIFY(K(John, x), K(x, Jane))=FAIL，因为x不能同时取2个值
- 置换只能在谓词/谓词公式的个体之间进行，不能改变谓词本身

求合一置换的算法

- 分歧集（不一致集）：设W是一个非空谓词公式集合，从左至右逐个比较W中各公式的符号，如果在第i(i⩾1)个符号处W中各公式第一次出现分歧（不一致），即至少存在F_j和F_k在第i个符号处不一样，而在i之前各公式符号均一样)，则全体谓词公式的第i个符号构成了W的分歧集D
- 分歧集的出现处一定是谓词或项（包括变量）的开始处
- 求mgu(最一般合一置换)的前提条件是：把所有公式中的变量换成不同符号的变量(合一)之前，所有公式中的变量都不一样，否则合一失败

Unify算法（递归形式）

```
function UNIFY(x, y, θ) returns a substitution to make x and y identical
  inputs: x, a variable, constant, list, or compound expression
          y, a variable, constant, list, or compound expression
          θ, the substitution built up so far (optional, defaults to empty)

  if θ = failure then return failure
  else if x = y then return θ
  else if VARIABLE?(x) then return UNIFY-VAR(x, y, θ)
  else if VARIABLE?(y) then return UNIFY-VAR(y, x, θ)
  else if COMPOUND?(x) and COMPOUND?(y) then
      return UNIFY(x.ARGS, y.ARGS, UNIFY(x.OP, y.OP, θ))
  else if LIST?(x) and LIST?(y) then
      return UNIFY(x.REST, y.REST, UNIFY(x.FIRST, y.FIRST, θ))
  else return failure
```

初始值为∅

X是否为变量？

是否为复合形式(函数)？

是否为列表形式(多元变量)？

```
function UNIFY-VAR(var, x, θ) returns a substitution

  if {var/val} ∈ θ then return UNIFY(val, x, θ)
  else if {x/val} ∈ θ then return UNIFY(var, val, θ)
  else if OCCUR-CHECK?(var, x) then return failure
  else return add {var/x} to θ
```

检查之前θ中是否已存在合一？

返回到主程序进行后续判断

项中包含变量

用x(此时为常量)置换var

Unify算法合一置换：例子

- {P(x), Q(u, v), R(f(a))}, {P(b), Q(c, d), R(f(y))}
- 设合一元素集合已确定：{x, (u, v), f(y)}, {b, (c, d), f(a)}
- 依次取<x ; b>, <u,v ; c,d>, <f(y) ; f(a)>
- 第1次合一：x/b
- 第2次合一：list—先后u/c、v/d
- 第3次合一：处理 op 部分，二者相同；然后arg(个体)部分，y/a

推理规则

- 对于原子公式p_i, p'_i, q，存在置换θ，使得对于所有的i都有

$$\theta(p_i) = \theta(p'_i)$$

则 $$\frac{p'_1,\cdots,p'_k \quad p_1 \wedge \cdots \wedge p_k \rightarrow q}{q}$$

- 上述公式说明，如果谓词公式$p'_1 \sim p'_k$与对应的谓词公式$p_1 \sim p_k$能够实现合一置换，那么规则$p_1 \wedge \ldots \wedge p_k \rightarrow q$对于$p'_1 \sim p'_k$也成立，即从前提$p'_1 \sim p'_k$可以推出结论$q$

推理常用定律

- ✓ 蕴涵等价律（蕴含到析取）

$$P(x) \rightarrow Q(x) \equiv \neg P(x) \vee Q(x)$$

- ◘ 狄摩根定律

$$\neg(P(x) \vee Q(x)) \equiv \neg P(x) \wedge \neg Q(x)$$

$$\neg(P(x) \wedge Q(x)) \equiv \neg P(x) \vee \neg Q(x)$$

- ◘ 分配律

$P(x) \vee (Q(x) \wedge R(x)) \equiv (P(\mathrm{x}) \vee Q(x)) \wedge (P(\mathrm{x}) \vee R(x))$　　同样有∧的分配

- ◘ 归谬律

$$(P(x) \rightarrow Q(x)) \wedge (P(x) \rightarrow \neg Q(x)) \equiv \neg P(x)$$

- ◘ ……

1.2.2 前向链接和后向链接算法

- 前向链接算法：正向推理—从前提推出结论
- 后向链接算法：反向推理—从结论倒推到前提

正向推理过程

★ 正向推理方法—前向链接算法

□ 前向链接算法的推理过程：

(1) 从已知事实即知识库中的原子公式开始，依次对知识库中的推理规则(以子句的形式出现)进行置换，检查规则前提部分的文字是否全部与知识库中的事实相匹配（约束）

(2) 如果是匹配的，则把该条规则已经做过置换的结论部分添加到知识库中，如果这个结论和需要推导出的结论（也称为查询）一致，则推理结束，结论获得证明

(3) 否则回到(1)，直到获得证明；或者再没有新的事实(推出的结论)加入，此时推理以证明失败结束

前向链接算法

- Input: 知识库KB，目标Goal α，置换θ(初始为空)
- Output:置换（证明成功）或FAIL（证明失败）
- 过程：局部变量*new*，用于存放每次迭代产生的新规则

1. **repeat until** *new* 为空
2. *new*←{}
3. **for** 每个KB中规则*r* **do**
4. **for** 每个使得$\theta(p_1 \wedge \cdots \wedge p_n) = \theta(p'_1 \wedge \cdots \wedge p'_n)$的置换θ，得到$\theta(q) = q'$
5. **if** q'是满足约束的新规则 **then do**
6. 将q'加入*new*
7. Σ←UNIFY(q', α)　（即对q'和α调用合一算法）
8. **if** Σ返回不是FAIL **then return** Σ
9. 将新规则*new*加入KB
10. **return** FAIL　★

反向推理过程

★ 反向推理方法—反向链接算法

□ 反向链接算法的推理过程：

(1) 将待证明的公式放入目标列表中，答案列表为空；

(2) 选取列表中的第一个目标，在知识库中寻找子句的头(即结论部分)能与目标合一的每个子句；每个子句创建了一个递归调用过程，在这个递归调用过程中子句的前提(子句的体)被放入到目标列表当中；

(3) 如果在知识库中无法找到与当前列表中任何目标进行匹配（合一）的事实，则证明失败，返回FAIL；

(4) 如果列表中的某个目标得到匹配（合一），即移出列表，其置换放入答案列表；如果目标列表为空，则证明成功，返回答案列表。

反向链接算法

- 算法3 反向链接算法Backward
- 输入：知识库（事实库）KB—有限子句集合，待证目标集合goals（因为该算法是一个递归过程，所以每次调用的输入可能不止一个待匹配的目标子句），当前置换θ（初始调用时为空）
- 输出：置换（证明成功）或FAIL（证明失败）
- 过程：局部变量*answers* (初始为空)，置换集合θ（初始为空）

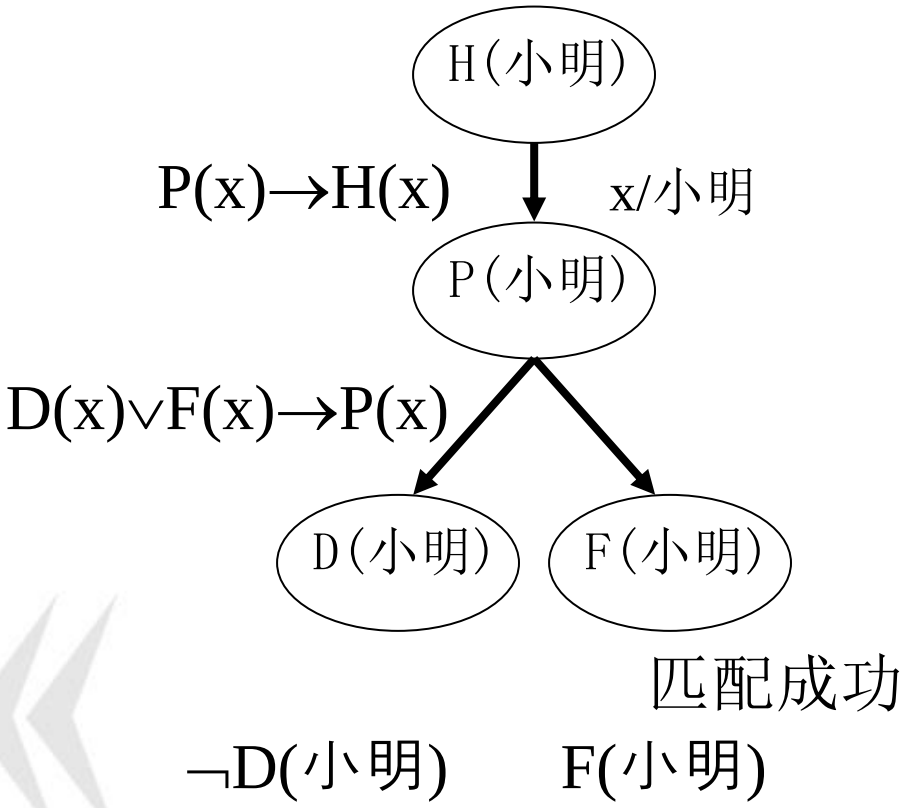
1. **if** goals为空 **then return** {θ}
2. **for each** 知识库KB中规则 r
3. $q' = head(r), q \in goals$
4. **if** $\theta' = Unify(q, q')$成功 **then** 将q移出$goals, body(r)$加入$goals$
5. $answers = Backward(KB, goals, \theta' \circ \theta) \cup answers$ (递归)
6. **return** FAIL ★

例1：快乐的小明

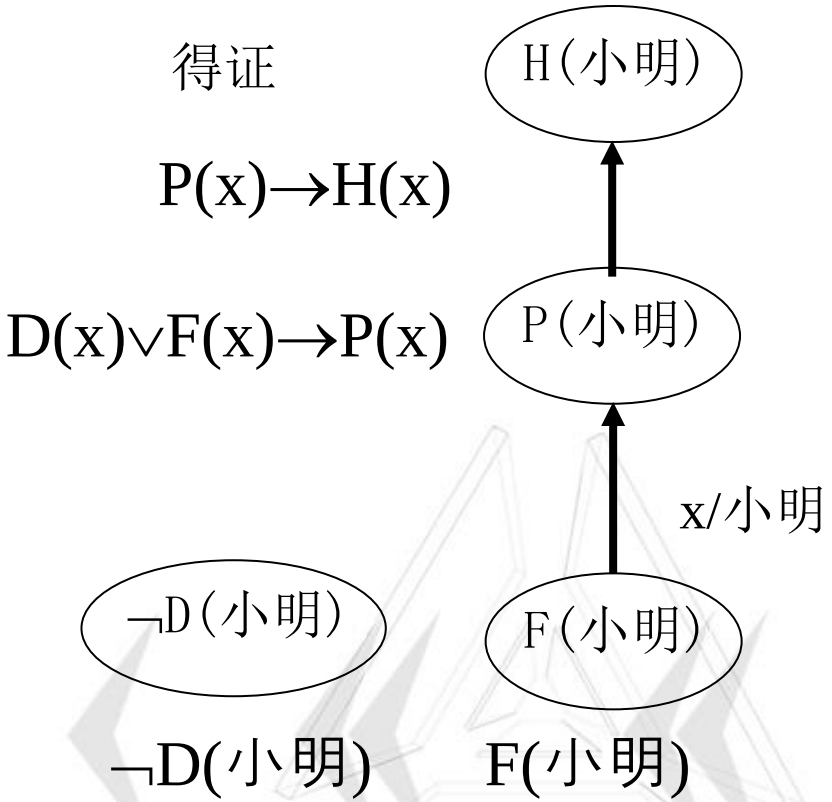
- 问题：小明不是一个勤奋的学生，但是他很幸运。凡是勤奋的或者幸运的学生都能通过考试，通过考试的学生都很快乐。证明：小明是快乐的。
- 谓词公式：
  - S(x)—x是学生，D(x)—x是勤奋的
  - F(x)—x是幸运的，$D(x) \vee F(x) \rightarrow P(x)$—勤奋或者幸运都能通过考试
  - $P(x) \rightarrow H(x)$—通过考试的学生是快乐的
  - ¬D(小明), F(小明)
  - 证明：H(小明)

快乐的小明：证明树

反向



正向



例2：猜帽子颜色游戏

- 游戏：有3顶红帽子、2顶蓝帽子，分别戴在甲乙丙3人头上，带帽前3人不知自己的帽子颜色，但知道帽子数量和颜色。
- 现场有一位主持人提问：谁知自己头上帽子颜色？每问一次，信息量都会增加，各人通过看到别人头上帽子颜色及分析，可以判定自己头上的帽子颜色。
- 问题：试用谓词公式表示游戏现场的问答过程
- 表示：用“红(x)”表示红帽子戴在x头上，“蓝(x)”表示蓝帽子戴在x头上—x∈{甲, 乙, 丙}
(y, z)也是同一论域

猜帽子颜色的游戏现场分析

- 现场有2顶蓝帽子：
- 蓝(x)∧蓝(y) ∧红(z)→问(1)∧答(z)
- 现场有1顶蓝帽子：
  - 蓝(x)∧红(y) ∧红(z)→问(1)∧¬答(x) ∧¬答(y) ∧¬答(z)
  - 蓝(x)∧红(y) ∧红(z)→问(2)∧ (答(y) ∨ 答(z))
- 现场没有蓝帽子：
  - 红(x)∧红(y) ∧红(z)→问(1)∧¬答(x) ∧¬答(y) ∧¬答(z)
  - 红(x)∧红(y) ∧红(z)→问(2)∧¬答(x) ∧¬答(y) ∧¬答(z)
  - 红(x)∧红(y) ∧红(z)→问(3) ∧ (答(x)∨答(y)∨ 答(z))
- 以上3种情况均覆盖 ★

例3：5人认证问题

- 前提：1 有五栋五种颜色的房子；　2 每一位房子的主人国籍都不同；3 这五个人每人只喝一种饮料，只抽一种牌子的香烟，只养一种宠物；4 没有人有相同的宠物，抽相同牌子的香烟，喝相同的饮料
- 条件：1 英国人住在红房子里　　2 瑞典人养了一条狗
  3 丹麦人喝茶　　4 绿房子在白房子左边
  5 绿房子主人喝咖啡　　6 抽万宝路香烟的人养了一只鸟
  7 黄房子主人抽三五牌香烟　　8 住在中间那间房子的人喝牛奶
  9 挪威人住第一间房子
  10 抽骆驼牌香烟的人住在养猫人的旁边
  11 养马人住在抽三五牌香烟的人旁边
  12 抽云斯顿香的人喝啤酒　　13 德国人抽健牌香烟
  14 挪威人住在蓝房子旁边
  15 抽骆驼牌香烟的人的邻居喝矿泉水
- 问题是：谁养鱼？？？

按照条件进行逻辑分析与填空

逻辑分析：依据前提4条和条件规则15条，一一进行约束、排除冲突；用过的规则即移除；首先确定房子颜色顺序

顺序	1	2	3	4	5
房子颜色	剩余	⑭	①	④	④
国籍	挪威人⑨		英国人①		
饮品			牛奶⑧	咖啡⑤	
吸烟牌子					
宠物					

互斥

按照条件进行填空：结果

挪威	丹麦	英国	德国	瑞典
				
				
三五	骆驼	万宝路	健牌	云斯顿

问题的谓词公式表示

- 5个外国人：国籍N的论域={英国佬, 瑞典佬, 丹麦佬, 挪威佬, 德国佬}
- 5人住房颜色：C={红, 绿, 白, 黄, 蓝}
- 5人宠物：P={狗, 鸟, 猫, 马, 鱼}
- 5人喝什么：D={茶, 咖啡, 牛奶, 啤酒, 矿泉水}
- 5人抽什么：S={万宝路, 三五, 骆驼, 云斯顿, 健牌}
- 描述：
  - N(英国)∧C(红)　　N(瑞典)∧P(狗)
  - N(丹麦)∧D(茶)　　left(C(白))=C(绿)　函数left
  - C(绿) ∧D(咖啡)　　S(万宝路) ∧P(鸟)

谓词公式表示

- 描述（续）：
  - C(黄)∧S(三五)　　seq(3) ∧D(牛奶)　　seq函数表示房屋次序(依次从左到右)
  - N(挪威) ∧seq(1)　　left(P(猫))=S(骆驼) ∨ right(P(猫)) = S(骆驼)
  - left(P(马))=S(三五)∨right(P(马))=S(三五)
  - S(云斯顿) ∧ D(啤酒)　　N(德国) ∧S(健牌)
  - right(N(挪威))=C(蓝)
  - left(S(骆驼))=D(矿泉水) ∨ right(S(骆驼))=D(矿泉水)
- 推理过程：将所有条件公式按照5个人进行一一归位　★

例4：医院评价

- 某报登出了国内20家大医院的名单，名单按它们在近3年中病人死亡率的高低排序。但实际情况是：（1）有些医院，留病人住院的时间长，病人死亡率因此就较高；有些医院，往往较早地动员患绝症而救治无望的病人出院，病人死亡率因此就较低。（2）这些医院中有2家老人医院和3家儿童医院。（3）在20家医院中有2家是肿瘤医院。因此，专家指出不能把名单上医院排列顺序作为评价这些医院医疗水平的一个标准。
- 试用逻辑推理方法支持专家的观点。
- 专家观点：以下谓词公式不成立
- $Hpt(x) \wedge dEad(x, 低) \rightarrow Rank(x, 高)$

谓词公式表示

- 直觉：条件(实际情况)部分没有给出死亡率与医院水平高低之间的关系
- 将所有事实(实际情况)列为谓词公式
  - T(y, x)住院时间，论域 $x \in D_T$={长, 中, 短}
  - H(y)医院类型，论域 $y \in D_H$={普通, 肿瘤, 老年, 儿童}
  - E(y, z)死亡率，论域 $z \in D_E$={低, 中, 高}
  - R(y, u)医院水平，论域 $u \in D_R$={低, 中, 高}
- 显然(依据统计数据或常识)，有以下公式：
- T(y, 长)→E(y, 高)　　T(y, 短)→E(y, 低)
- H(老年)→E(老年, 高)　　H(肿瘤)→E(肿瘤, 高)

推证过程

- 专家观点：Hpt(x)∧dEad(x, 低)→Rank(x, 高)不成立
- 公式变形：¬ (H(y)∧E(y, 低)) ∨ R(y, 高) ⇒
  (狄摩根律) ¬H(y) ∨ ¬E(y, 低) ∨ R(y, 高)
- 要上式不成立，需要析取的3个谓词均不成立
  - R(y, 高)不在事实中，不成立
  - ¬E(y, 低)不在事实中，不成立
  - ¬H(y)不存在(非医院)，不成立
- 因此，原公式不成立 ★

1.2.3 谓词公式向子句转化

- 为逻辑推理的反证法做准备：要证明谓词公式A为真，就是要证明¬A为假—在语义上要证明：对于任何论域上的任何赋值，¬A恒为假
- 任何论域、任何赋值（数学上的无穷）在实际证明中很难实现，所以需要寻找有限个论域和赋值

✓ Herbrand提出：从所有解释当中选出一种有代表性的解释，并严格证明一旦公式在代表性解释（H解释）中为假，则在所有解释中为假（赫勃兰定理）

Herbrand (赫勃兰)定理

- 保证这样的反证推理方法正确性的定理称为Herbrand（赫勃兰）定理：
- 找到了代表性的解释，即公式A(无∃前束范式)是不可满足的，当且仅当A(通过子句集)在所有H解释下都为假
- H解释可以通过公式（子句集）的语义树形象地说明
- 所以需要将谓词公式A转化为子句集，并给出H解释

谓词公式的形式

- 合取范式：形如$A = A_1 \wedge A_2 \wedge \ldots \wedge A_n$的公式，其中$A_1$~$A_n$均为子句（只含有析取连接词的公式）
- 前束范式：形如$(Q_1 x_1 \ \ldots \ Q_n x_n) M(x_1 \ldots x_n)$的合取范式，M中不再含有量词，Q是量词（全称量词、存在量词）
- Skolem（斯科伦）标准形：在前束范式中消去存在量词后得到的公式
- 公式A对应的子句集：将Skolem标准形中的全称量词去掉，将合取连接词去掉而使用逗号代替，从而形成一个子句集S（因为经过前束范式、 Skolem标准形变换之后A_1~A_n仍然是子句）

消去存在量词

■ 消去存在量词的步骤：

（1）若存在量词不在任何全称量词之后，则公式中被存在量词量化的变量以某个不同于公式中任何其他常量名字的常量c代替，并消去存在量词

（2）若存在量词在k个全称量词之后，则公式中被存在量词量化的变量用被前k个全称量词量化的变量$x_1, \ldots, x_k$的某个函数$f(x_1, \ldots, x_k)$的形式代替，f的名字不同于公式中任何其他函数的名字，但对函数形式没有要求；然后消去存在量词；函数f称为Skolem函数

谓词公式转化为子句的步骤

■ 公式A化为子句集S的步骤（共9步）：

(1) 消去等价和蕴含符号：蕴含转化为析取；　　(2) 将否定符号转移到每个谓词之前：应用狄摩根定律（即$\neg(A \wedge B) = \neg A \vee \neg B$）；

(3) 变量标准化：约束变量各不相同；　　(4) 消去存在量词：应用消去存在量词步骤（见上），得到Skolem标准形；

(5) 公式化为前束型：全部全称量词移到公式的最前面，得到的两部分称为前缀和母式；　　(6) 母式化为合取范式：外层连接符全部是合取，里层连接符全部为析取；

(7) 去掉所有全称量词；　　(8) 母式化为子句集：每个合取项间的合取符号(∧)用逗号代替，即得子句集；

(9) 子句变量标准化：让每个子句中的变量各不相同

子句转化举例

- 例5：凡是在凯德广场购物的人都有机会获得一份奖品
- H(x)是人，B(x, y) x在y购物，M(y)购物广场y
- G(x, y, z) x在y获得礼品z，F(z)是礼品
- $\forall x (H(x) \wedge B(x, Kaide) \wedge M(Kaide) \rightarrow \exists u H(u) \wedge G(u, Kaide, z) \wedge F(z))$
- 变换：$\forall x (\neg(H(x) \wedge B(x, Kaide) \wedge M(Kaide)) \vee (\exists u H(u) \wedge G(u, Kaide, z) \wedge F(z)))$
- 前半部分用狄摩根定律：$\neg H(x) \vee \neg B(x, Kaide) \vee \neg M(Kaide)$

转化过程

- 后半部分对∃u应用Skolem函数：$u = f(x)$
- H(f(x))∧ G(f(x), Kaide, z) ∧ F(z)
- 整个公式就是：
- (¬H(x)∨¬B(x, y)∨¬M(Kaide))∨(H(f(x))∧ G(f(x), Kaide, z) ∧ F(z))
- 分配律可得3项子句：
- ¬H(x)∨¬B(x, Kaide)∨¬M(Kaide)∨H(f(x)),
  ¬H(x)∨¬B(x, Kaide)∨¬M(Kaide)∨G(f(x), Kaide, z),
  ¬H(x)∨¬B(x, Kaide)∨¬M(Kaide)∨F(z) ★

归结规则

- 公式转换为子句集的目的：下一步进行互补文字（即原子公式与其否定式）的消去
- 归结规则：两个互补文字消去称为归结
- 单元归结规则：一个子句和一个文字进行互补文字消去
  如果$m = \neg l_i$，则有

$$\frac{l_1 \vee \cdots \vee l_k \quad m}{l_1 \vee \cdots \vee l_{i-1} \vee l_{i+1} \vee \cdots \vee l_k}$$

- 全归结规则：如果两个子句中包含互补文字，即$m_j = \neg l_i$，则有

$$\frac{l_1 \vee \cdots \vee l_k \quad m_1 \vee \cdots \vee m_n}{l_1 \vee \cdots \vee l_{i-1} \vee l_{i+1} \vee \cdots \vee l_k \vee m_1 \vee \cdots \vee m_{j-1} \vee m_{j+1} \vee \cdots \vee m_n}$$

归结反演定理(S定理)

- 归结反演推理过程：如果$A \to B$成立，则B也为真；我们用证明子句集$S = \{A \to B, \neg B\}$为假也即子句集S是不可满足的来证明$A \to B$成立
- 子句集不可满足：经过一系列变换后得到最终子句集为空，在变换过程使用归结规则
- 参与推理的各个公式（前提+结论的否定）转换为子句集，然后逐步对消子句集合中的互补文字（即L和┑L）而最终得到一个空子句 □ / 推理过程使用了归结规则和反证法，所以称为归结反演方法（消解法）
- S定理：给定公式A及相应的子句集S，则A是不可满足的当且仅当S是不可满足的

赫勃兰论域

- Herbrand论域(H论域)：设S为子句集，H_0是子句所包含的全体常量集，若S中子句不含常量，则任选一常量a，令$H_0 = \{a\}$；对于i≥1，令$H_i = H_{i-1} \cup \{f(t_1, \cdots, t_n) | n \geq 1\}$，$f$是S中的所有函数符号，$t_1 \sim t_n$是$H_{i-1}$中的所有元素；$H_\infty = \cup_{i \geq 0} H_i$
  - H_∞称为 H论域，H_i称为S的 i 阶常量集，集合中元素称为H论域的基项
  - Herbrand基原子集（H基）：设S为子句集，H_∞是H论域，P是S中的所有原子公式（不含否定词），则$\overline{H} = \{P(t_1, \cdots, t_n) | t_i \in H_\infty\}$称为S的基原子集，其中每个元素称为基原子；这是S中所有原子公式在H论域上所有可能取值的集合

子句集转化为基原子集例子

- 例6：求子句集 $S=\{P(x), Q(f(a)) \vee \neg R(g(b))\}$的基原子集
- 对于S_2有：$H_0=\{a,b\}$，$H_1=\{a,b\}$
  $\cup\{f(a), f(b), g(a), g(b)\}=\{a,b,f(a),f(b),g(a),g(b)\}$
- $H_2=\{a,b,f(a),f(b),g(a),g(b),f(f(a)),f(g(a)),\cdots\}$
- $H_\infty=\{a,b\}\cup\{f(u),g(v)|u,v\in H_\infty\}$
  （此处是一个递归表示）
- S中有3个原子公式P、Q、R，2个函数f、g和2个常量a、b，组合可得S的基原子集：
- $\overline{H}=$
  $\{P(a),Q(a),R(a),P(b),Q(b),R(b),P(f(a)),Q(f(a)),\cdots\}$

★

赫勃兰解释

- Herbrand解释：即子句集S的H解释，是一种语义结构，由H论域、常量和变量取值、函数和谓词映射组成
- 同时确定$\overline{H} = \overline{H_1} \cup \overline{H_2}$，其中$\overline{H_1}$中原子公式取真值，$\overline{H_2}$中原子公式取假值，通常给定一个即可
- H解释的直观表示：引入关于H解释的语义树方法
- 语义树表示：要寻找子句集 S的 H解释的不可满足性质，可以把所有 H解释 (即H基原子集) 展现在一棵语义树上，然后观察 S对应各个原子公式的真假值

子句集的语义树

- 子句集S的语义树：设子句集S，对应H基为$\overline{H}$，则 S的语义树ST定义如下：

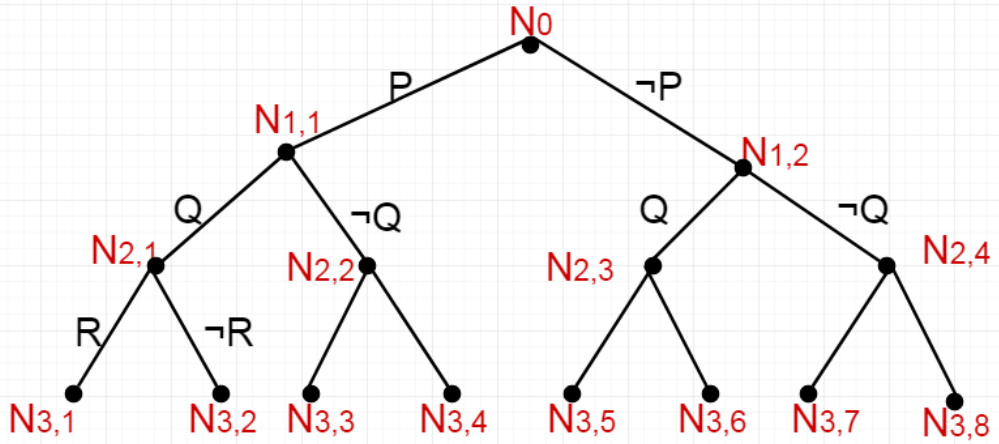
(1) ST是一棵树；

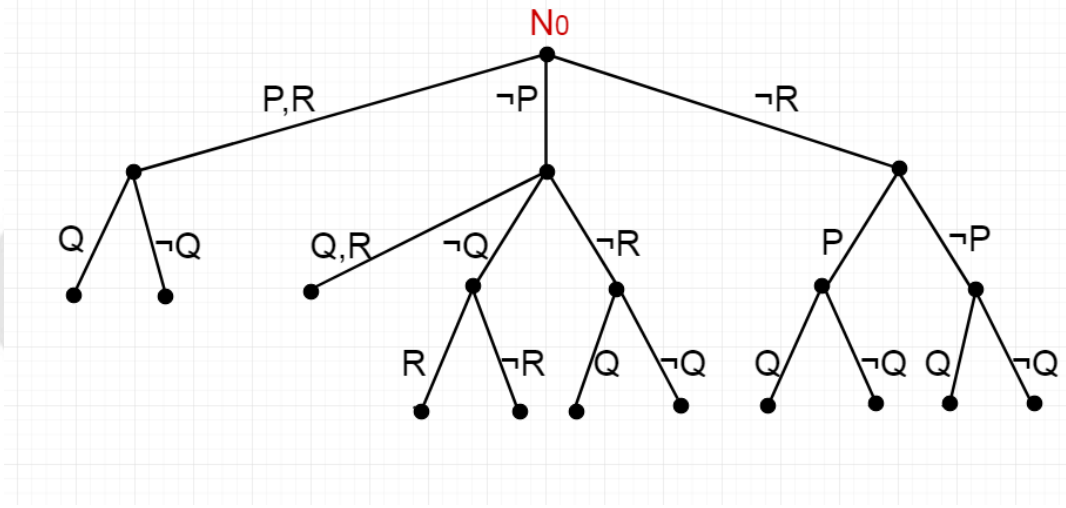
(2) ST的节点均不带标记，而边的标记是基原子且不包含其中的变量(即只有谓词公式符号；边的标记可能不止一个，用逗号分割)；

(3) 从每个非叶节点只生成有限个边L_1~L_n，令Q_i是每个L_i标记的合取，则$Q_1 \vee Q_2 \vee \cdots \vee Q_n$为永真（即总有一个$Q_i$为真）；

(4) 从根节点到任一叶子节点的路径上，所有边标记的并集中不含重复的基原子，也不含互补的基原子

语义树举例





$S = \{P(x), Q(x), R(x)\}$

H解释对应的语义树

$H_\infty = \{a\}$

H基原子集：$\overline{H} = \{P(a), Q(a), R(a)\}$

语义树形式

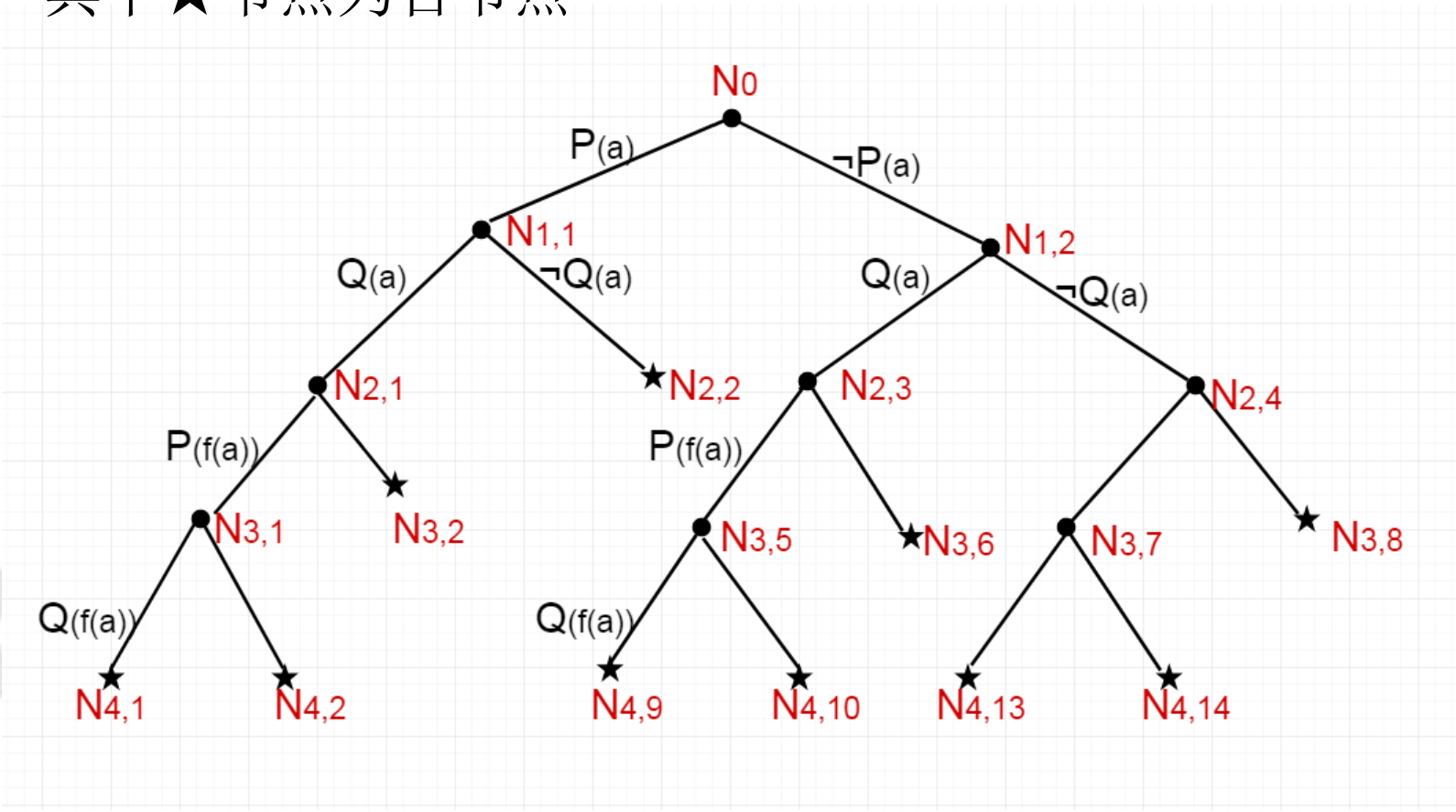
- 对于给定的子句集S，其对应的语义树ST一般不唯一，如上例
- 产生多个语义树的原因：虽然S的H基原子集相同，但是原子公式在 语义树 上可以有多种组合方式，因此语义树不唯一；对于子句集$S=\{P(x),Q(x),R(x)\}$，每个边上的原子公式有多种组合方式，如P、Q组合，P、R组合(上例中的下图)等等
- 规范语义树：如果一棵语义树每条边的标记均为一个原子公式(上例中的上图)
- 完备语义树：如果一棵规范语义树从根节点到任一叶节点的路径上所有边标记的并集中包含子句集中每个原子或其负原子，则该语义树称为完备的(完全的) / 上例中两图中第一个为完备语义树

语义树与H解释

- 基例：S的某个子句C中所有变量符号均以S的 H论域元素（常量）代入，所得子句C' 称为C的一个基例 （基原子）
- 否节点：如果语义树从根节点到节点N_{ij}的路径$I(N_{ij})$使S中某个子句的某个基例为假，而其父辈节点均为真，即$I(N_{ij})$是使基例为假的最小路径，则称N_{ij}为否节点(或失败节点)
- 封闭语义树：如果一棵完备语义树的每个分枝上都有一个否节点，则称为封闭语义树
- 语义树与 H解释之间关系：语义树的每个分支就是一个H解释
- 因为 H基是个无穷集合（前面例子是特例，为的是画语义树方便），所以其对应的语义树也是无限的；但是封闭语义树给出了因为否节点的存在，使得在反证意义下就有了有限解释

封闭语义树举例

- $S=\{\neg P(x)\vee Q(x), P\big(f(x)\big), \neg Q(f(x))\}$的封闭语义树；其中★节点为否节点



赫勃兰定理

- 在完备语义树的每一条路径上都有 S的全部原子(或负原子)，对这些原子公式的真值判定就能决定S的一个解释；而封闭语义树的存在，使得这种真值判断成为一种有限操作；所以从子句集不可满足的角度来说，对于无限延伸的完备语义树只需搜索到一个否节点（使S中某个子句为假），则该分支即被证明为不可满足；搜索操作是有限的
- Herbrand定理：子句集S不可满足，当且仅当它所对应的每个完备语义树均包含一个封闭语义树
- 也可表述为：S是不可满足的，当且仅当存在S的一个不可满足的有限基例集 (用于自动定理证明)

1.2.4 消解法的实施(消解策略)

- “子句集不可满足”的实施策略：消去一切可满足的子句，使子句集最后为空，此时子句集不可满足
- 删除策略：删去可满足的一切文字，包括：
  - 删除永真式
  - 删除单文字（只有一个原子公式的子句，只要设置为1 即可满足）
  - 删除单一符号的子句（即S中只有原子公式P而没有¬P，删去含P的子句，操作同上）
  - S中正负文字（P和¬P）的析取式即为永真式，所以可以成对消除；消除过程需要进行合一运算，以保证文字相同

删除可满足的文字

- 文字合并规则：如果子句C含有n(n>1)个相同的文字，那么删去其中的n-1个，结果以[C]表示
- 取因子后使用文字合并规则：如果子句C中的多个文字具有mgu σ，那么[Cσ]称为C的一个因子；取因子之后，子句C中可能出现相同的文字，根据文字合并规则可以删除重复部分
- 求二元消解式（如下）

- 变换成子句集的目的，就是要通过处理最简单的文字(正负原子公式)形式来完成判定

二元消解式

- 二元消解式：设C_1、C_2是无公共变量的子句，分别含文字L_1、L_2，而L_1和$\neg L_2$有mgu σ，则子句

$R(C_1, C_2) = [(C_1\sigma - L_1\sigma) \vee (C_2\sigma - L_2\sigma)]$称为$C_1$和$C_2$的二元消解式，其中$(C_1\sigma - L_1\sigma)$和$(C_2\sigma - L_2\sigma)$分别表示从$C_1$、$C_2$中删去$L_1$、$L_2$后剩余的部分；$L_1$、$L_2$称为被消解的文字，$C_1$、$C_2$称为父子句

- 子句的二元消解式包括以下4种：(1) C_1、C_2的二元消解式；(2) C_1的一个因子(即经过取因子处理)和C_2的二元消解式；(3) C_1和C_2的一个因子的二元消解式；(4) C_1一个因子和C_2一个因子的二元消解式
- 消解法就是消除子句集中的各种互补文字

二元消解式举例

- 例7：求子句$C = P(x,a) \vee P(b,y) \vee Q(x,y,c)$的消解式

取因子：子句C经过置换$\sigma = \{x/b, y/a\}$后可得

$[C\sigma] = [P(b,a) \vee P(b,a) \vee Q(b,a,c)] = P(b,a) \vee Q(b,a,c)$

- 设有子句$C_1 = P(x) \vee Q(x)$，$C_2 = \neg P(g(y)) \vee \neg Q(b) \vee R(z)$，求其二元消解式
- 如果采用置换$\sigma = \{x/g(y)\}$，
- 则$R(C_1, C_2) = Q(g(y)) \vee \neg Q(b) \vee R(z)$
- 如果采用置换$\theta = \{x/b\}$，
- 则$R(C_1, C_2) = P(b) \vee \neg P(g(y)) \vee R(z)$
- 注意：求二元消解式时，不能同时消去2对互补文字，即不能同时消去$P(x)$和$\neg P(x)$、$Q(x)$和$\neg Q(x)$，那不是C_1, C_2的逻辑推理结果

消解法合理性定理和完备性定理

- 子句C的推导：给定子句集S，如果存在一个有限的子句序列$C_1, C_2, \cdots, C_k$，使得每个C_i或者属于S，或者是$C_1 \sim C_k$某些子句的二元消解式，并且$C_k = C$，则称从S中可以推导出子句C；当C为空子句 □ 时，称该序列为S的否证过程；找出一个子句集S推出空子句 □ 的过程，则S就不可满足
- **合理性定理**：若子句集S是可满足的，则S推导出的任何一个子句也是可满足的，即不能推出空子句；其逆否形式就是：若子句集S推导出空子句 □ ，则S是不可满足的
- **完备性定理**：子句集S是不可满足的，当且仅当从S可推导出空子句 □
- 根据定理S可知，S对应公式 A→B 不可满足；与前提矛盾

消解法的理论证明过程

存在S消解为空的过程 } 完备性定理
←→存在有限封闭语义树
←→Herbrand解释不可满足 } Herbrand定理
←→子句集S不可满足 } 定理S
←→公式G=A∧¬B(¬(A→B))
[子句集合]是不可满足的 } 反证法
←→A→B是有效的

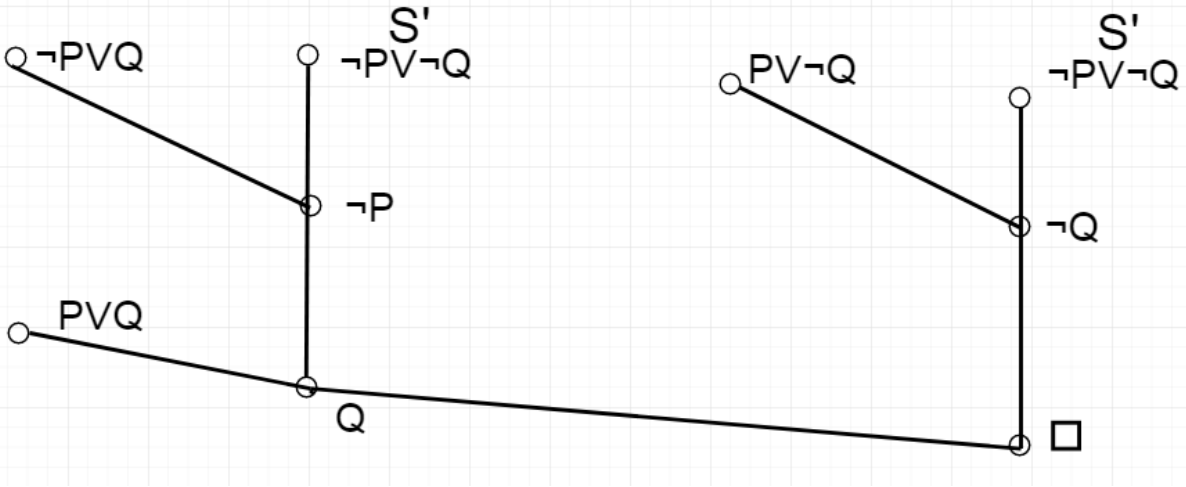
常用消解策略

★ 常用的消解策略

- 使用消解法进行谓词公式推理，根本目标是最终消去子句集中的全部子句，得到一个空集合；考虑2种手段：一个是删除无用子句；另一个是禁止产生无用子句
- 删除策略：即对整个子句集S进行检查，查找其中的永真式，将其删除；另外一种方式就是通过置换，使S中某些子句变换为与另外一些子句互补的形式，从而消去—但是检查往往很耗费时间
- 于是采用 2种禁止策略：一种是限制参加消解的子句，这里主要介绍此种方法；另一种是限制消解方式，如规定文字的消解次序

消解策略举例

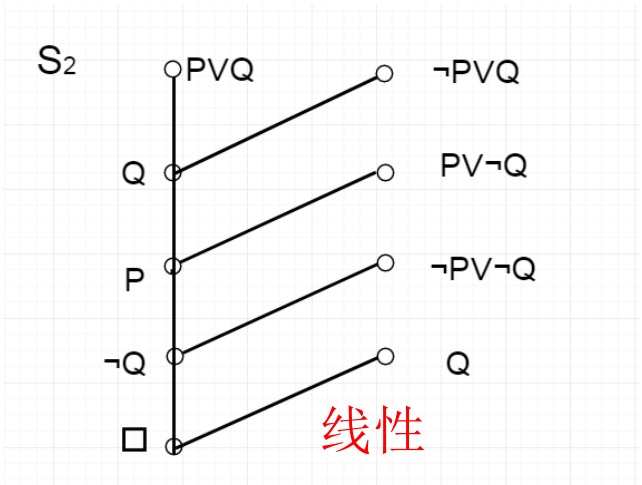
- 例8：设有子句集$S=\{P(x)\vee Q(x),\neg P(x)\vee Q(x),P(x)\vee\neg Q(x),\neg P(x)\vee\neg Q(x)\}$
- 则$S'=\{\neg P(x)\vee\neg Q(x)\}$，因为当$P(x)$和$Q(x)$都取真值时，$S-S'=\{P(x)\vee Q(x),\neg P(x)\vee Q(x),P(x)\vee\neg Q(x)\}$是可满足的
- 对于S的消解，分别取S'和$S-S'$作为消解双方(限制双方必须各来自一方)，消解得到 □ ★

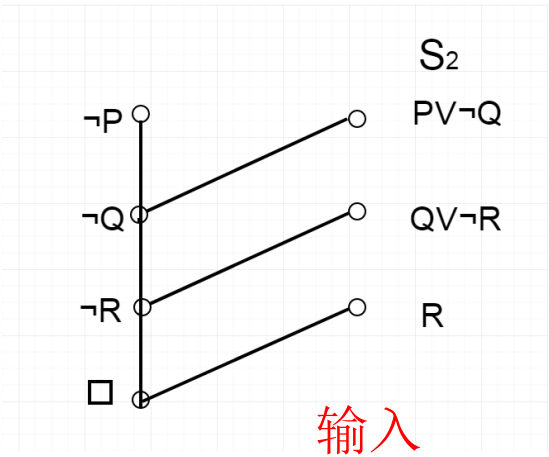


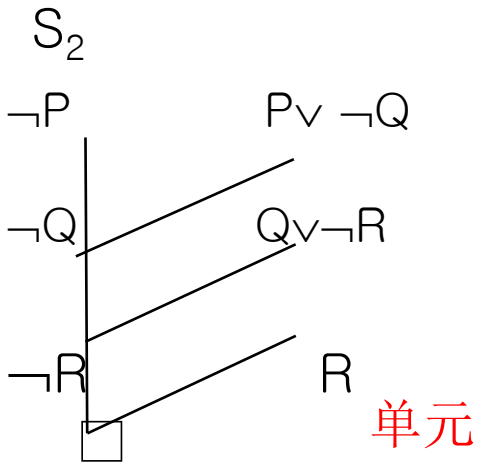
动态支持集消解策略

- 动态支持集策略包括：线性消解、输入消解和单元消解
  - 线性消解策略：在动态支持集消解定义中令$S_2^n = \{C_n\}$，则此策略称为线性消解策略；此时C_n称为消解中心子句，另一方称为边子句；选择初始消解中心子句很重要，否则可能得不到空子句
  - 输入消解策略：在动态支持集消解定义中令$S_2^n = S$，则此策略称为输入消解策略—因为每次消解必须有原子句集S中的子句参加(即输入子句)，由此得名
  - 单元消解策略：在动态支持集消解定义中令$S_2^n =$ {S^n中所有单子句和单因子}，则此策略称为单元消解策略—此策略中每次参加消解的一方总有一个是单文字，由此得名

消解策略举例





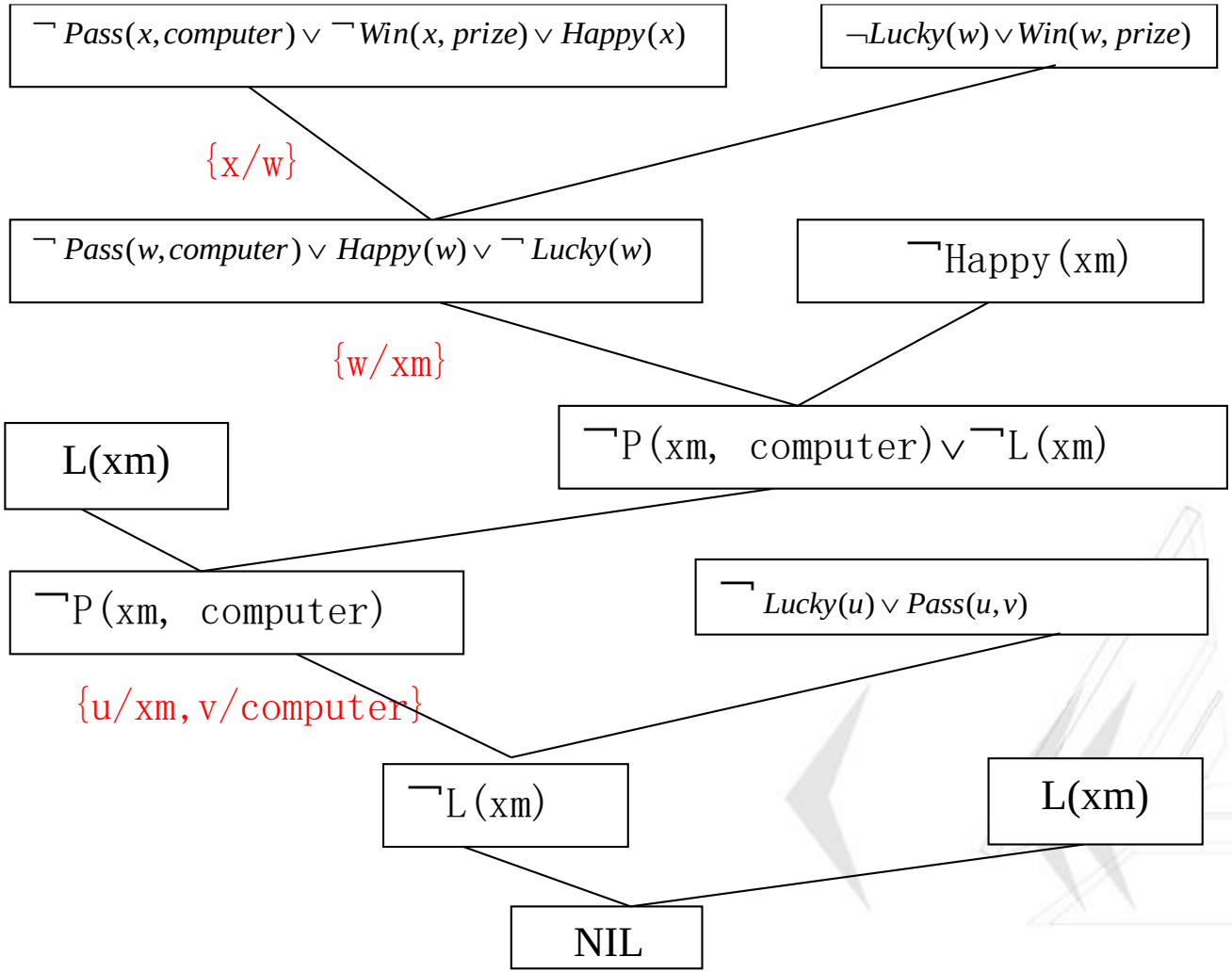


- 采用线性消解对子句集S=
  $\{P(x) \vee Q(x), \neg P(x) \vee Q(x), P(x) \vee \neg Q(x), \neg P(x) \vee \neg Q(x)\}$
  进行消解
- 试分别采用输入消解、单元消解2种策略对子句集$S = \{\neg P, P \vee \neg Q, Q \vee \neg R, R\}$来进行消解否证
- 这里集中关注消解过程，设变量全部置换为相同的项，故只保留谓词公式符号

例9：“快乐的小明”问题

- 任何通过计算机考试并获奖的人都是快乐的，任何肯学习或幸运的人都可以通过所有考试；小明不肯学习但他是幸运的，任何幸运的人都能获奖。求证：小明是快乐的
  - 任何通过计算机考试并获奖的人都是快乐的
  - $\forall x(Pass(x, computer\_test) \wedge Win(x, prize)) \rightarrow Happy(x)$
  - 任何肯学习或幸运的人都可以通过所有考试
  - $\forall x \forall y(Study(x) \vee Lucky(x)) \rightarrow Pass(x, y)$
  - 小明不肯学习但是他是幸运的 $\neg Study(xm) \wedge Lucky(xm)$
  - 任何幸运的人都能获奖：$\forall x(Lucky(x)) \rightarrow Win(x, prize)$
  - 小明是快乐的：$Happy(xm)$；其否定形式：$\neg Happy(xm)$

“快乐的小明”消解树


¬Pass(x,computer) ∨ ¬Win(x, prize) ∨ Happy(x)
¬Lucky(w) ∨ Win(w, prize)
{x/w}
¬Pass(w,computer) ∨ Happy(w) ∨ ¬Lucky(w)
¬Happy(xm)
{w/xm}
L(xm)
¬P(xm, computer) ∨ ¬L(xm)
¬P(xm, computer)
¬Lucky(u) ∨ Pass(u,v)
{u/xm,v/computer}
¬L(xm)
L(xm)
NIL

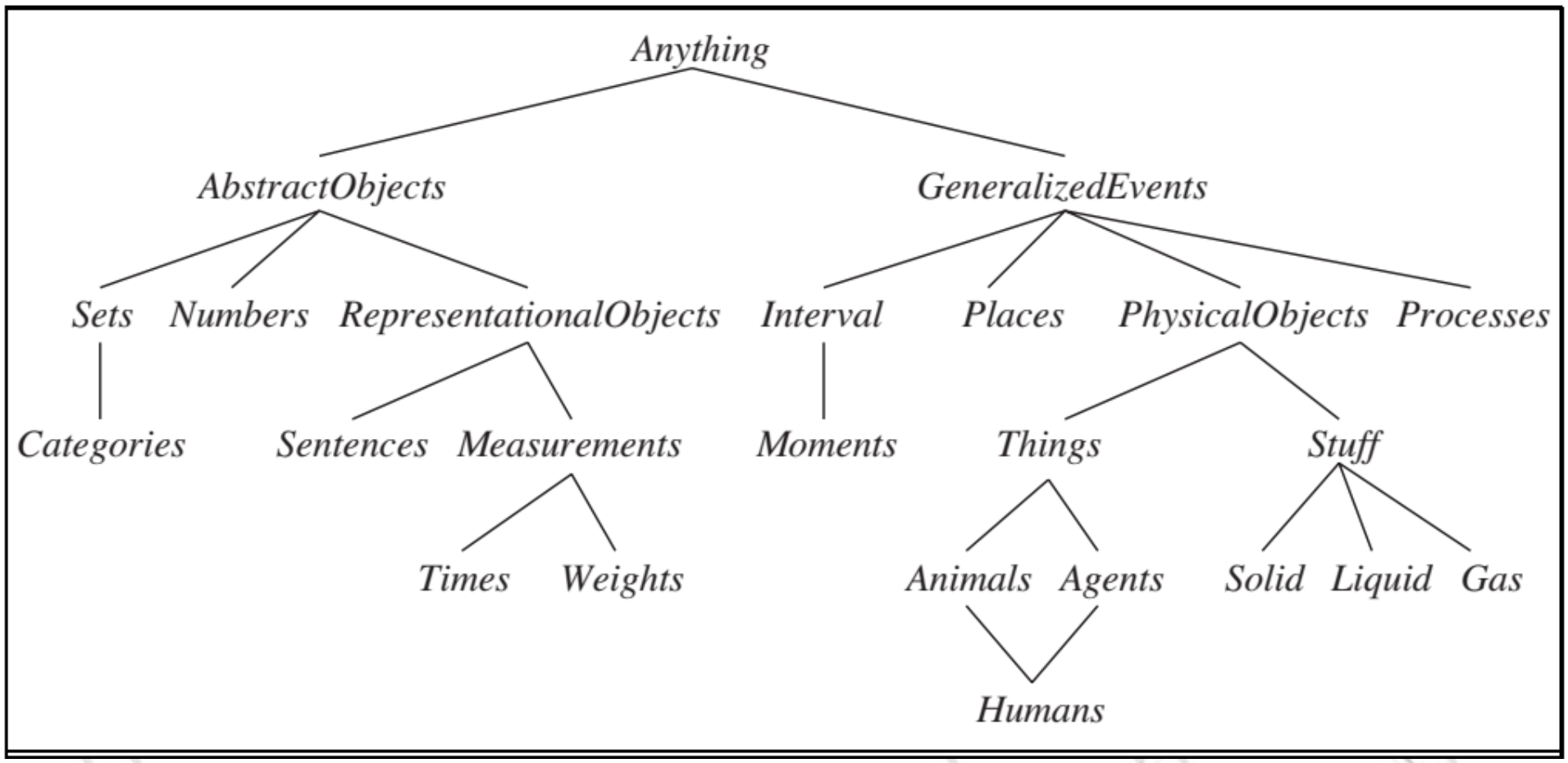
1.3 Logic Language for Real-World
用于真实世界表示的逻辑语言

- 1.3.1 真实世界(物理世界)的逻辑描述
- 1.3.2 事件的逻辑描述
- 1.3.3 网购事件的表示
- 1.3.4 基于逻辑规则的常识推理举例

1.3.1 真实世界的逻辑描述形式

- 表示世界上一切事物的前景是令人生畏的(AIMA第3版中译本p361)
- 真实世界需要描述哪些方面？一些通用概念
  - ✓事件
  - ✓时间
  - ✓物理对象(实体)
  - ✓信念(认识)
- 本体论工程—上位本体论
- 用途：或多或少应用于专用领域，统一不同领域的概念—有限的成功

一种上位本体论


Anything
AbstractObjects
GeneralizedEvents
Sets
Numbers
RepresentationalObjects
Interval
Places
PhysicalObjects
Processes
Categories
Sentences
Measurements
Moments
Things
Stuff
Times
Weights
Animals
Agents
Solid
Liquid
Gas
Humans

物理世界表示：类别和对象

- 大规模分类系统应用非常广泛，如行业分类、产品分类；分类层次：类别→子类→成员/对象，具有继承关系

- 一个对象是一个类别的成员。

$$BB_9 \in Basketballs$$

- 一个类别是另一个类别的子类。

$$Basketballs \subset Balls$$

- 一个类别中的所有成员拥有某些属性。

$$(x \in Basketballs) \Rightarrow Spherical(x)$$

球形的

- 一个类别的成员可以通过某些属性来识别。

$$Orange(x) \wedge Round(x) \wedge Diameter(x) = 9.5" \wedge x \in Balls \Rightarrow x \in Basketballs$$

- 一个类别作为整体拥有某些属性。蓝球直径约24.6厘米(9.5英寸)

$$Dogs \in DomesticatedSpecies$$

驯化物种

物理世界表示：部分

- 物理世界的对象表示为Part of (部分)层次结构

$PartOf(Bucharest, Romania)$

$PartOf(Romania, EasternEurope)$

$PartOf(EasternEurope, Europe)$

$PartOf(Europe, Earth)$

PartOf 关系是传递的和自反的；即

$$PartOf(x, y) \land PartOf(y, z) \Rightarrow PartOf(x, z)$$

$$PartOf(x, x)$$

因此，我们可以得出结论 $PartOf(Bucharest, Earth)$。

物理世界表示：复合对象

- 通过各部分关系结构刻画一个复合对象

两足动物身体上有两条腿：

$$Biped(a) \Rightarrow \exists l_1, l_2, b \quad Leg(l_1) \wedge Leg(l_2) \wedge Body(b)$$
$$\wedge part\ Of(l_1, a) \wedge part\ Of(l_2, a) \wedge part\ Of(b, a)$$
$$\wedge Attached(l_1, b) \wedge Attached(l_2, b)$$
$$\wedge l_1 \neq l_2 \wedge [\forall l_3 \quad Leg(l_3) \wedge part\ Of(l_3, a) \Rightarrow (l_3 = l_1 \vee l_3 = l_2)]$$

两足动物 a，腿 l，躯干 b，附着attached
如果存在第3条腿的话，也是两条腿之中的某一条

- 定义没有结构的复合对象“束bunchof”，以区别于某种对象的集合

$$\forall x \quad x \in s \Rightarrow PartOf(x, BunchOf(s))$$
$$\forall y\ [\forall x \quad x \in s \Rightarrow PartOf(x, y)] \Rightarrow PartOf(BunchOf(s), y)$$

物理世界表示：度量

$$Length(L_1) = Inches(1.5) = Centimeters(3.81)$$

单位之间的转换用某个单位的倍数等于另一个单位来完成：

$$Centimeters(2.54 \times d) = Inches(d)$$

可以为磅和千克、秒和天、元和分写出类似的公理。量度可以像下面那样用来描述对象：

$$Diameter(Basketball_{12}) = Inches(9.5)$$

$$ListPrice(Basketball_{12}) = \$(19)$$

$$d \in Days \Rightarrow Duration(d) = Hours(24)$$

- 度量的排序

$$e_1 \in Exercises \wedge e_2 \in Exercises \wedge Wrote(Norvig, e_1) \wedge Wrote(Russell, e_2) \Rightarrow Difficulty(e_1) > Difficulty(e_2)$$

$$e_1 \in Exercises \wedge e_2 \in Exercises \wedge Difficulty(e_1) > Difficulty(e_2) \Rightarrow ExpectedScore(e_1) < ExpectedScore(e_2)$$

物理世界表示：物质

- 物体和物质：可数名词与不可数名词

任何一个黄油对象的部分也是一个黄油对象：

$$x \in Butter \ \wedge \ partOf(y, x) \ \Rightarrow \ y \in Butter$$

现在可以说黄油在 30 摄氏度时熔化：

$$x \in Butter \ \Rightarrow \ MeltingPoint(x, Centigrade(30))$$

- 物质的属性：固有属性vs非固有属性
  - 固有属性：属于对象每个实体，即使对象被分割（如黄油切成两半），每个部分仍然具有其固有属性，如密度、熔点、口味等
  - 非固有属性：如重量、长度、形状等不能保持不变
- 只包含固有属性的实体是物质，包含了非固有属性的实体就是物体；stuff不指定固有属性，Thing不指定非固有属性

1.3.2 事件的逻辑描述

- 事件具有过程性，专门的描述方法—事件演算(Event Calculus)
- 事件events: 某个时间点/段上发生的事实(如某人从某地飞某地)
- 流fluents: 某个对象(实体)在某段时间内所处状态(如某人在某地)
- 事件演算 使 事件和流 具体化

$$E_1 \in Flyings \ \wedge \ Flyer(E_1, Shankar) \ \wedge Origin(E_1, SF) \ \wedge Destination(E_1, DC)$$

$$E_1 \in Flyings\ (Shankar, SF, DC)$$

事件演算谓词集

$T(f,t)$	流 f 在时间 t 为真
$Happens(e,i)$	事件 e 发生在时间区间 i
$Initiates(e,f,t)$	事件 e 使流 f 从时间 t 开始成立（启动）
$Terminates(e,f,t)$	事件 e 使流 f 从时间 t 开始不再成立（终结）
$Clipped(f,i)$	流 f 在时间区间 i 的某个时间点开始不再为真（剪切）
$Restored(f,i)$	流 f 在时间区间 i 的某个时间点开始变为真（恢复）

将 T 扩展到时间区间也很方便：一个流在一个时间区间上成立，如果它在这个区间里的每个点都成立：

$$T(f,(t_1,t_2)) \quad \Leftrightarrow \quad [\forall t\ (t_1 \leqslant t < t_2) \Rightarrow T(f,t)]$$

事件过程中的暂停和恢复

$$Happens(e,(t_1,t_2)) \wedge Initiates(e,f,t_1) \wedge \neg Clipped(f,(t_1,t)) \wedge t_1<t \Rightarrow T(f,t)$$

持续发生中

$$Happens(e,(t_1,t_2)) \wedge Terminates(e,f,t_1) \wedge \neg Restored(f,(t_1,t)) \wedge t_1<t \Rightarrow \neg T(f,t)$$

持续暂停中

其中 *Clipped* 和 *Restored* 定义为

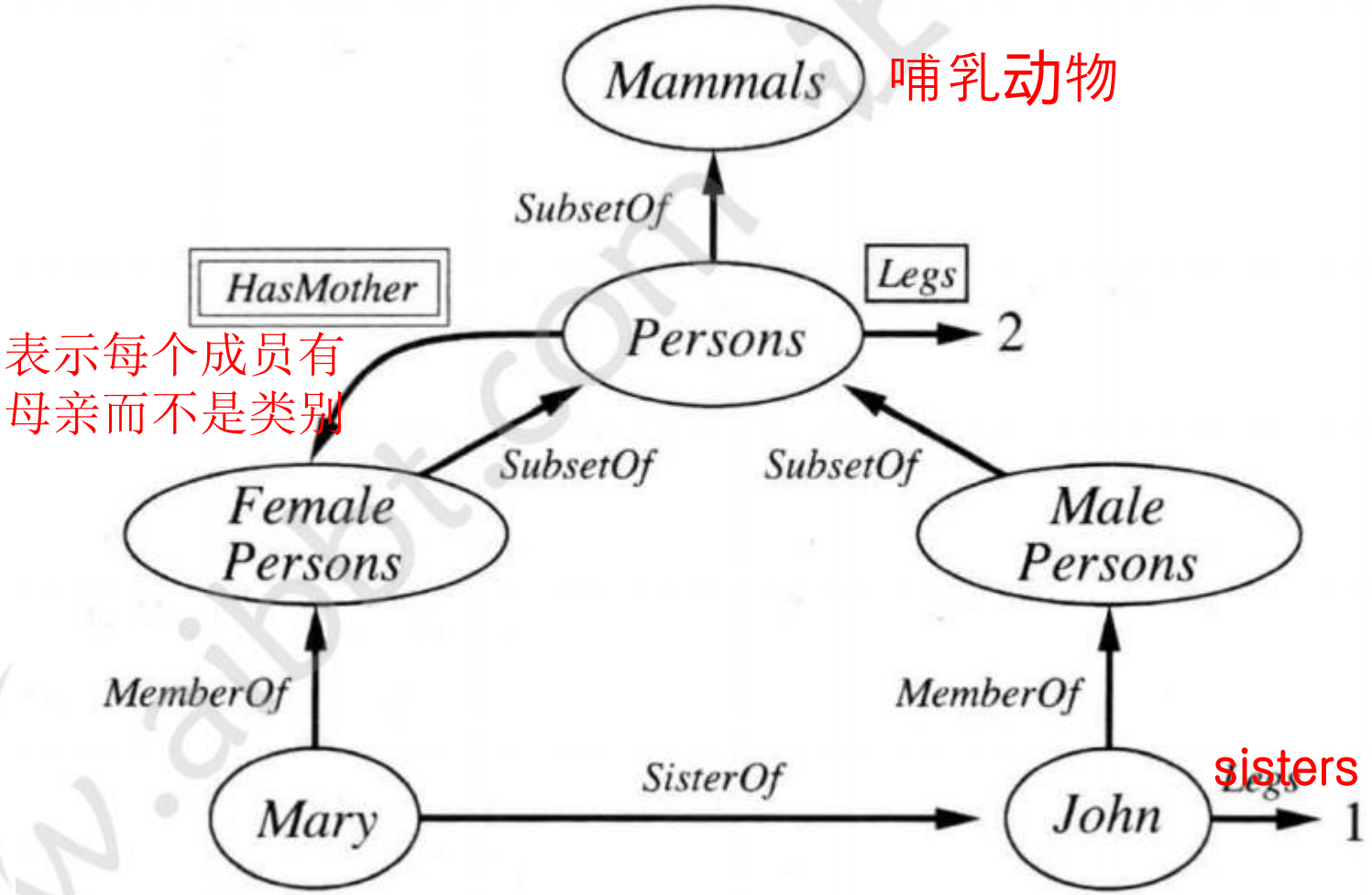
暂停 $Clipped(f,(t_1,t_2)) \Leftrightarrow \exists e,t,t_3 \quad Happens(e,(t,t_3)) \wedge t_1 \leqslant t<t_2 \wedge Terminates(e,f,t)$

恢复 $Restored(f,(t_1,t_2)) \Leftrightarrow \exists e,t,t_3 \quad Happens(e,(t,t_3)) \wedge t_1 \leqslant t<t_2 \wedge Initiates(e,f,t)$

例10：小明乘机旅行

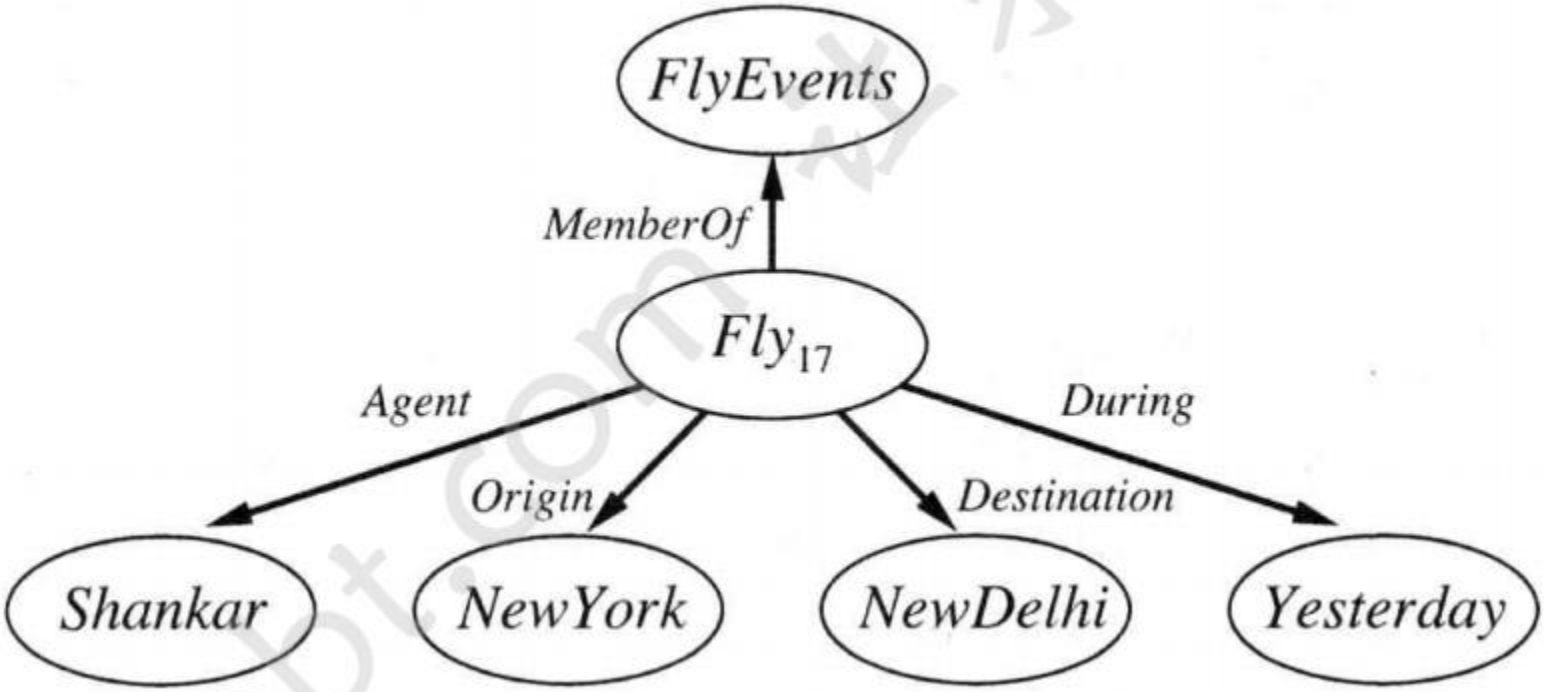
- 飞行过程：小明从深圳飞到了哈尔滨
- 描述：e1∈Flying　　t1< t2< t3< t4∈time
  At∈f(流)　　xm小明∈Person (Entity)
  SZ深圳, HRB哈尔滨∈Place (Entity)
- 流：At(xm, SZ)　　At(xm, HRB)
- T(At(xm, SZ), t1)　　小明t1时在深圳
- Happens(Flying, t2, t3)　　在t2~t3期间飞行
- Initiates(Flying, ¬At(xm, HRB), t2)
- Terminates(Flying, ¬At(xm, SZ), t3)
- T(At(xm, HRB), t4)　　小明t4时在哈尔滨

类别推理系统


Mammals
哺乳动物
SubsetOf
HasMother
Legs
Persons
2
表示每个成员有
母亲而不是类别
SubsetOf
SubsetOf
Female Persons
Male Persons
MemberOf
MemberOf
SisterOf
sisters
Mary
John
1

类别推理扩展

- 将二元关系扩展到多元关系



描述逻辑语言

CLASSIC 语言（Borgida 等人，1989）是一种典型的描述逻辑。图 12.7 显示了 CLASSIC 描述的语法 [1]。例如，要说单身汉是未结婚的成年男性，我们写为：

Bachelor = *And*(*Unmarried*, *Adult*, *Male*)

Concept	→	**Thing** | *ConceptName*
	|	**And**(*Concept*, ···)
	|	**All**(*RoleName*. *Concept*)
	|	**AtLeast**(*Integer*, *RoleName*)
	|	**AtMost**(*Integer*, *RoleName*)
	|	**Fills**(*RoleName*, *IndividualName*, ···)
	|	**SameAs**(*Path*, *Path*)
	|	**OneOf**(*IndividualName*, ···)
Path	→	[*RoleName*, ···]

图 12.7　CLASSIC 语言一个子集的描述语法

1.3.3 网购事件的表示

- 例11：商店与商品：链接与分类

Example Online Store

Select from our fine line of products:

- Computers
- Cameras
- Books
- Videos
- Music

页面呈现形式

```
<h1>Example Online Store</h1>
<i>Select</i> from our fine line of products:
<ul>
<li> <a href="http://example.com/compu">Computers</a>
<li> <a href="http://example.com/camer">Cameras</a>
<li> <a href="http://example.com/books">Books</a>
<li> <a href="http://example.com/video">Videos</a>
<li> <a href="http://example.com/music">Music</a>
</ul>
```

实现上述页面的HTML语句

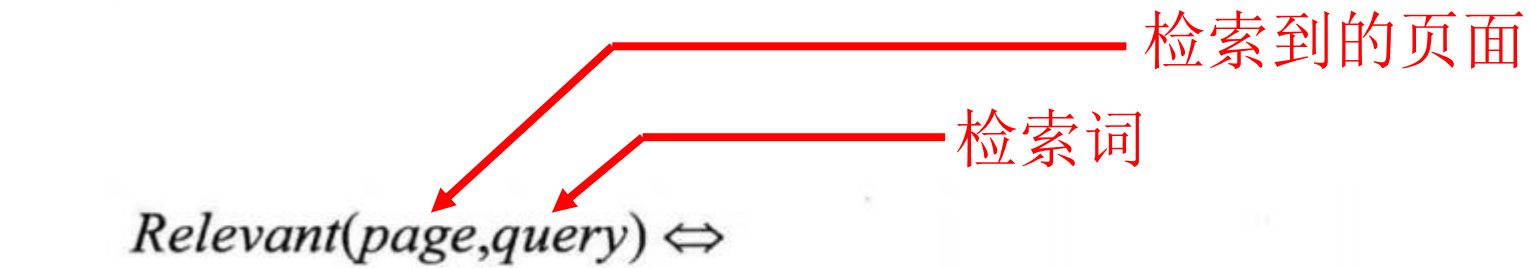
链接：从主页到指定页面

- Agent关于网店的知识

$$Amazon \in OnlineStores \wedge Homepage(Amazon, "amazon.com")$$
$$Ebay \in OnlineStores \wedge Homepage(Ebay, "ebay.com")$$
$$ExampleStore \in OnlineStores \wedge Homepage(ExampleStore, "example.com")$$

- 网店主页给出数码产品的主要类别（如上页图），包括计算机、照相机等
- 次要类别在主类给出链接的页面，这个页面可以通过检索到达

检索商品



$$Relevant(page,query) \Leftrightarrow$$
$$\exists store,home \quad store \in OnlineStores \wedge Homepage(store, home)$$
$$\wedge \exists url, url_2 \quad RelevantChain(home, url_2, query) \wedge Link(url_2, url)$$
$$\wedge page = Contents(url)$$

- 从当前页面home(实际上也是一个url网址)出发，通过网页地址url2到网址url的超链接，到达页面page，其中包含了query相关内容

相关链接 (同类词检索)

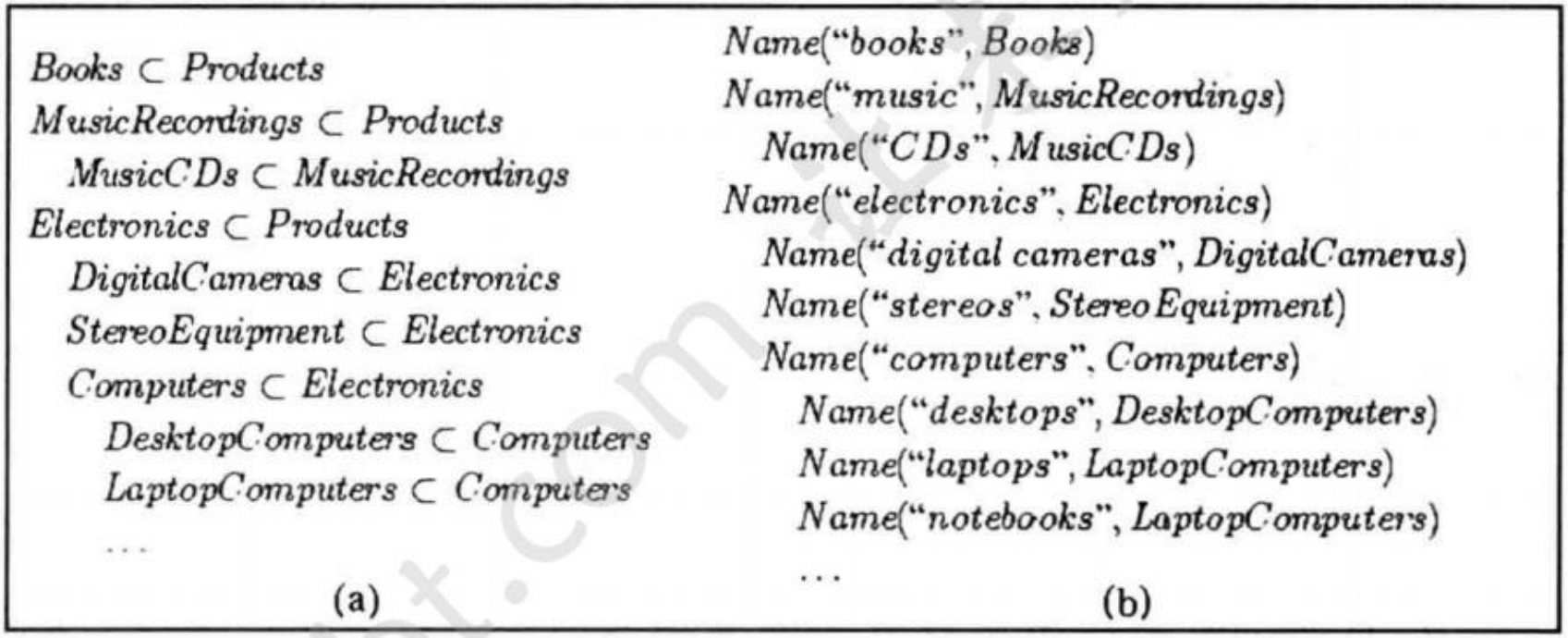
$$RelevantChain(start, end, query) \Leftrightarrow (start = end)$$
$$\vee(\exists u, text \quad LinkText(start, u, text) \wedge RelevantCategoryName(query, text)$$
$$\wedge RelevantChain(u, end, query))$$

- 通过检索词query从初始页面start检索到end页面，所得结果
- 或者为空start = end；或者存在一个网址u，该网址中含有text的内容，text应该是与query同类的商品名称，或者是这一类别的超类或者是其子类；且存在网址 u 到目标网址end的链接 (甚至可能u = end)

$$RelevantCategoryName(query, text) \Leftrightarrow$$
$$\exists c_1, c_2 \quad Name(query, c_1) \wedge Name(text, c_2) \wedge (c_1 \subseteq c_2 \vee c_2 \subseteq c_1)$$

相关商品名

- *text* 和 *query* 命名同一个类别——例如，“notebooks”和“laptops”。
- *text* 命名一个像“computers”这样的超类。
- *text* 命名一个像“ultralight notebooks（超轻笔记本电脑）”这样的子类。



Books ⊂ *Products*
MusicRecordings ⊂ *Products*
  MusicCDs ⊂ *MusicRecordings*
Electronics ⊂ *Products*
  DigitalCameras ⊂ *Electronics*
  StereoEquipment ⊂ *Electronics*
  Computers ⊂ *Electronics*
    DesktopComputers ⊂ *Computers*
    LaptopComputers ⊂ *Computers*
    ...

(a)

Name(“*books*”, *Books*)
Name(“*music*”, *MusicRecordings*)
  Name(“*CDs*”, *MusicCDs*)
Name(“*electronics*”, *Electronics*)
  Name(“*digital cameras*”, *DigitalCameras*)
  Name(“*stereos*”, *StereoEquipment*)
  Name(“*computers*”, *Computers*)
    Name(“*desktops*”, *DesktopComputers*)
    Name(“*laptops*”, *LaptopComputers*)
    Name(“*notebooks*”, *LaptopComputers*)
    ...

(b)

- 左侧为类别与子类；右侧为商品名与其类别

商品描述

- 被描述的一件具体商品的信息

```
IBM ThinkBook  970.   Our Price:   $399.00
```

后面跟着各种技术规格说明，我们希望一个封装器提取类似下列信息：

$$\exists c, offer \quad c \in LaptopComputers \wedge offer \in ProductOffers \wedge$$
$$Manufacturer(c, IBM) \wedge Model(c, ThinkBook970) \wedge$$
$$ScreenSize(c, Inches(14)) \wedge ScreenType(c, ColorLCD) \wedge$$
$$MemorySize(c, Gigabytes(2)) \wedge CPUSpeed(c, GHz(1.2)) \wedge$$
$$Offered\ Product\ (offer, c) \wedge Store\ (offer, GenStore) \wedge$$
$$URL(offer, \text{“example.com/computers/34356.html”}) \wedge$$
$$Price(offer, \$(399)) \wedge Date(offer, Today)$$

对商品的比较

- 设有以下3款笔记本电脑

A：1.4GHz CPU，2GB RAM，250GB disk，$299。
B：1.2GHz CPU，4GB RAM，350GB disk，$500。
C：1.2GHz CPU，2GB RAM，250GB disk，$399。

- 查询以下规格商品：
- d∈LaptopComputers ∧ ScreenSize(d, Inches(14)) ∧ MemorySize(d, Gigabytes(2)) ∧ CPUSpeed(d, GHz(1.2)) ∧ OfferedProduct(offer, d) ∧ Price(d, $(300))
- 某种查询语句能够比较上述要求与封装器提取的商品规格描述（见上页）；显然，C款（IBM ThinkBook 970）笔记本电脑无法满足要求，但是A款可以满足

1.3.4 基于逻辑规则的常识推理举例

- 常识知识
  - 常识知识就是一个普通人应该知道的知识，当然常识知识的范围是不易确定的，与教育情况、自然社会环境等都相关
  - AI研究者普遍认为如果没有常识知识，一个AI系统将很难完成理解、推理等任务，因此常识知识对于AI系统是非常重要的
  - 常识知识在日常生活中往往是隐式使用的，没有显式地表示出来，通常是以自然语言形式表现出来，在人类交流时当做共享的已知前提
  - 但是，对于AI系统来说，没有合适的显式表示方式，就意味着机器无法使用；所以必须将这些自然语言形式的常识以逻辑符号形式明确表示出来，再进行推理

常识知识分类（1）

- 常识知识包括：

常识可以大致分为三类（包括但不限于）*：Social commonsense（社交常识），Temporal commonsense（时间常识）、Physical commonsense（物理常识）

Social commonsense

Social commonsense 是指人们对于推测他人的心里和精神状态的能力，比如推测别人的动机、推测别人下一步要做什么，等等

另外，social commonsense 还指的是一套公认的，可接受的社会行为标准，比如：评价别人的体重是一个不礼貌的行为；这些常识都隐含在我们日常的行为和决定，但是对于机器来说，需要明确地教给他们

* 知乎： https://zhuanlan.zhihu.com/p/350110729

常识知识分类（2）

Temporal commonsense

一般来说，自然语言中很少包含明确的时间信息，人们获取时间信息一般从模糊的语言信息和常识中推断出来

比如：从“张教授正在度假”这句话中，我们就能知道我们与张教授不能很快见面，但是，从“张教授正在散步”这句话中，我们就会知道，我们也许可以和张教授很快见到面

这种推断就需要知道“正在”“during”的常识，即“taking a walk”和“taking a vacation”一般所需要花的时间（minutes 和 days）

还有其他时间相关的常识，比如：次数、顺序、频率等等（可参考数据集 MC-TACO）MC-TACO：dataset of 13k question-answer pairs that require temporal commonsense comprehension

常识知识分类（3）

Physical commonsense

Physical commonsense 包括日常使用的物体objects的物理性质和功能可见性（affordances ）

比如：一个玻璃杯摔倒地上，我们就能知道它会被摔碎。（可参考数据集PIQA: Reasoning about Physical Commonsense in Natural Language）

- 此外，还有百科知识中简单的一部分也可看做是常识知识

基于符号逻辑的常识推理

- 以某种显式方式表示常识知识，是知识工程关注的主要对象；区别于自然语言推理
- 这里举出一个例子，即ConceptNet，这是MIT媒体实验室自1999年开始的一个项目OMCS（Open Mind Common Sense）所构建的知识库，并且具有多语言版
- ConceptNet给出了以下20种事物之间的关系，可以视作常识知识应该具备的表示形式的一部分（参见维基百科）：

常识知识的谓词表示

（1）IsA （2）UsedFor （3）HasA （4）CapableOf
（5）Desire （6）CreatedBy (例子："cake" can be created by "baking")
（7）PartOf （8）Causes （9）LocateNear （10）DefinedAs
（11）AtLocation (例子：a "cook" can be at a "restaurant")
（12）SymbolOf (例子：X represents Y) （13）MadeOf
（14）ReceivesAction (例子："cake" can be "eaten")
（15）HasPrerequisite (例子：X can't do Y unless A does B)
（16）MotivatedByGoal (例子：You would "bake" because you want to "eat")
（17）CausesDesire (例子："baking" makes you want to "follow recipe")
（18）HasSubevent (例子："eat" has subevent "swallow")
（19）HasFirstSubevent (例子：The first thing required when you're doing X is for entity Y to do Z)
（20）HasLastSubevent

常识推理问题

- 常识推理是一个基于常识知识的、取得某个场景特定方面信息后对该场景其他方面做出推断的过程(Mueller, 2015)
- 下面举例说明什么是常识推理：

(1) 小明在一间大学教室里自习，他把一本英汉词典放在桌上；当他去卫生间再回来时，发现词典不见了。请问：词典哪里去了？常识推理结果：被别人拿走了

(2) 一只饥饿的猫盯着桌上的鱼，它跳到桌子旁的椅子上。请问：这只猫下一步的行动是什么？常识推理结果：跳到桌子上去吃鱼

例12：常识推理问题—猫吃鱼

- 给定句子“一只饥饿的猫盯着桌上的鱼，它跳到桌子旁的椅子上”，试用常识推理对饥饿的猫下一步行动作出预测，可以采用类似ConceptNet的表示方式定义与行动相关的描述
- 下面给出本句子涉及的常识表示：
- 实体相关：IsA(猫, #动物)，IsA(鱼, #食物)，Status(猫, 饥饿)，AtLocation(鱼, 桌子上)，AtLocation(猫, 椅子上)，LocateNear(桌子, 椅子)，Location(桌子上)，Location(椅子上)
- 动作/事件相关：Action(#动物, 跳)

猫吃鱼—常识推理过程（1）

- 推理规则：

①Status(#动物, 饥饿)→Desire(#动物, #食物)

②AtLocation(#动物, #位置1)∧Action(#动物, 跳)→AtLocation(#动物, #位置2)

③Desire(#动物, #食物)∧AtLocation(#食物, #位置)∧AtLocation(#动物, #位置)→Eat(#动物, #食物)

- 其中“#号”表示类别；IsA表示类别的实例化，即可以将实例代入类别

猫吃鱼—常识推理过程（2）

- 由此可得该问题的常识推理过程如下：
  - IsA实例化使“猫”和“鱼”均可使用相应的推理规则
  - 由猫的“状态”通过规则①可得：Desire(猫, 鱼)
  - 调用规则③，尚不满足条件；继续查找其他规则
  - 调用规则②可得：AtLocation(猫, 桌子上)
  - 此时规则③条件均得到满足，可得Eat(猫, 鱼)
- 根据匹配的3条规则，可以得出推理结果：猫想要鱼，猫从椅子上跳到桌子上，猫在桌子上吃鱼；简言之：猫要从椅子上跳到桌子上吃鱼 ★
- 上述推理过程遵循的就是一个正向匹配过程（即正向链接算法），每个知识描述都以类似谓词逻辑的形式表示；如何把一个句子转换为类谓词逻辑的形式，可能需要自然语言理解（或分析）技术，也是最困难的部分

本章小结

- 复习：一阶谓词逻辑系统
- 一阶谓词逻辑推理算法：前向链接算法、后向链接算法
- 推理例子（推理算法的不同方式应用）
- 子句的转化与归结反演方法（消解法）
- 对于真实世界的逻辑描述形式（实体、时间、事件等）
- 事件演算、查询表示
- 基于逻辑的常识推理简介

参考文献

- AIMA 第8、9、12章
- 陆汝钤 编著，人工智能（下册），科学出版社，1996年
- E. T. Mueller, Commonsense Reasoning (2nd Edition), Morgan Kaufmann, 2015