第三部分　知识的表示和推理

第13章　命题演算

13.1　对特征值加以约束

前面已经论述了两种截然不同的、为一个 agent世界建模的方法：基于图标的和基于特征的方法。二元值特征是对这个世界的描述——什么是真的，什么是假的。而图标表示则是对这个世界的某些方面的模拟。虽然模拟比描述更直接，因而也常常更有效，但是描述有它自身的一些优势。特征值容易与别的 agent进行通信，而图标模型就很难分解为独立的部分来满足增量的通信。特征值的计算与图标模型的构建和修改相比，常常只需要较少精细的知觉处理。并且，当不能直接感知某些特征值时，可以利用加在某个 agent所处的世界的约束，从其他的特征值推断出。

进一步说，一个agent环境的某些信息是很难或者说是不可能由图标来表述的。例如：

- 普遍规律，如“所有的蓝色盒子都是可推动的”。
- 否定信息，如“积木A不在地板上（没有说积木A在哪儿）”。
- 不确定信息，如“或者积木A在积木B上面，或者积木A在积木C上面”。

有些这种难以表述的信息可以用公式表示为对特征值的约束，这些表示某个 agent所处世界的重要知识的约束常常被用来推断那些不能被直接感知的特征值。推断有关一个 agent当前（*present*）状态的信息（使用基于特征中约束的计算），可以与从一个agent行为的将来（*future*）状态的计算（使用本书第二部分讨论过的搜索方法）作对照。第一个方法称为“推理（*reasoning*）”，第二个称为“投影（*projecting*）”。在以下几章中，先不考虑投影的问题，而集中讨论推理。

包含有关特征值的推理有几项应用。可以确信，当 agent（甚至是反应型agent）决定行动的时候，推理能增强它们的效力。但是，也存在许多别的应用。例如：已经可以验证用合适的特征集来表述各种物理系统的功能，包括生物的和电子机械的等。这些特征中的约束把物理的或别的与有机体或器械相关的规律进行编码，然后可以在其他的目标中进行推理，用来诊断这些系统中的故障。例如，与“原因”相联系的特征可以从与“症状”相联系的特征推断出。这种方法是人工智能应用中的重要一类——专家系统（*expert system*）的基础。

用一个富有启发性的例子来开始推理技术的讨论。设想一个能举起一块积木的机器人，假如那个木块是可举的（即不是太重的），并且假设这个机器人有足够的电池能源。假如以上条件都满足，那么当机器人试着举起这个它所握住的木块时，它的手臂就移动。可以用二元值特征来表述这些不同的条件：

x_1 (BAT_OK) （电池能源合适）

x_2 $(LIFTABLE)$ （可举的）

x_3 (*MOVES*) （移动）

用这些便于记忆的特征名称（放在括号中）可有助于讨论。假设机器人能感知 *BAT_OK*（通过读量表）和*MOVES*（通过联角传感器），但不能感知*LIFTABLE*。但知道*LIFTABLE*的值对机器人完成其必须完成的任务来说是很重要的。从上面的描述，我们知道假如 *BAT_OK*和*LIFTABLE*的值都为1，那么*MOVES*也如此。所以，假如当这个机器人试着要移动这块积木时 *MOVES*值为0，那么我们知道或者 *BAT_OK*或者*LILFTABLE*（或者两者）肯定值为0。如果*BAT_OK*被感知到值为1，那么*LIFTABLE*的值一定为0。既然我们能如此推理，那么机器人也可如此。所需要的是一种能用于表达特征中的约束和特征值的语言 (*language*)和能进行必要推理的推理(*inference*）机制。而命题演算（*propositional calculus*）作为布尔代数的一种延伸，为此提供了必要的工具。

上面例子中的约束可用命题演算语言如下表示：

BAT_OK∧LIFTABLE   MOVES

其中，∧的意思是“合取”，而 的意思是“蕴含”。

与这种语言相联系的机制可以用来从这种语句（ *statement*）中推出结论（*consequence*)。既然逻辑语言在人工智能中是如此重要，那么，必须在表述基于使用这些语言的更复杂和更具智能的agent之前，对这些语言作更详细的说明。

首先是一些定义。逻辑包含

- 一种语言（具有一个句法用以规定在这种语言中什么是合法的表达式)。
- 推理规则 用以操作这种语言中的语句。
- 语义 用以把这种语言中的要素和某些主题（*subject matter*）中的要素联系起来。

我们要学习两种逻辑语言：第1种是两种语言中较为简单的，叫做命题演算；第2种，也是更有用的，叫做一阶谓词演算（*first-order predicate calculus*， *FOPC*)。由于许多在一阶谓词演算中重要的概念可以更为简单地在命题演算中作介绍，因此首先介绍命题演算。

13.2 语言

下面从形式上描述命题演算中的组成元素。此刻最好不要把某种意义与这种语言的组成元素联系起来，把我们现在正在做的想像为对一个无意义的游戏规则的描述，以后我们再谈论意义。以下是这种语言的组成元素：

原子（*atom*)：两个最明显的原子是T和F，还有可数的以大写字母开头的那些字符串的无限集合，如：P, Q, R, ..., P_1, P_2, ON_A_B, 等等。

联结词（ *connective*)：∨、∧、 和¬，分别称为：“析取（ *disjunction*)”、“合取(*conjunction*)”、“蕴含（*implication*)”和“非（*negation*)”(以后我们会给出与这些名称相关的联结词的意义，而现在它们只是无意义的符号)。

合式公式（*well-formed formulas*，*wff*)（也称为句子*sentence*）的语法：

- 任何原子都是一个合式公式。

例如：P, R, p3

- 假如ω_1和ω_2 是合式公式，那么以下这些也是：

$\omega_1 \vee \omega_2$ （ω_1和ω_2的析取）

$\omega_1 \wedge \omega_2$ （ω_1和ω_2的合取）

ω_1 ω_2 （蕴含）

$\neg\omega_1$ （ω_1的非）

原子以及前面带有 ¬ 符号的原子叫做文字（*literal*）。在ω_1 ω_2中，ω_1被称为蕴含的前项（*antecedent*），而ω_2被称为蕴含的后项(*consequent*)。

合式公式的例子：

$(P \wedge Q) \supset \neg P$

$P \supset \neg P$

$P \vee P \supset P$

$(P \supset Q) \supset (\neg Q \supset \neg P)$

$\neg\neg P$

• 不再有别的合式公式。

例如：P ¬ 不是一个合式公式。

注意以上例子中有语言外的分隔符“（”和“）”的使用。它们根据递归定义把合式公式组成次级（sub）合式公式。有些逻辑处理把这两个分隔符公式化为语言的组成部分，然而在此用得有点宽松并且较直观。合式公式可以通过递归地使用它们的定义来识别。例如，（P ∧ Q） ¬R是一个合式公式。首先，P和Q是合式公式，所以（P ∧ Q）是合式公式；并且由于R是一个合式公式，¬R也是一个合式公式，因而，（P ∧ Q） ¬R是一个合式公式。

13.3 推理规则

从一些合式公式推出另一些合式公式可以有许多方法，称它们为推理规则（ *rule of inference*）。一条推理规则的典型形式是：γ可以从α（或从α 和 β）推出。下面是一些通用的推理规则：

• 合式公式ω_2可以根据合式公式 ω_1和ω_1 ω_2推出（称之为假言推理或演绎推理（ *modus ponens*））。

• 合式公式$\omega_1 \wedge \omega_2$ 可以根据两个合式公式ω_1和ω_2推出（∧引入）

• 合式公式 $\omega_2 \wedge \omega_1$可以根据合式公式$\omega_1 \wedge \omega_2$ 推出（∧交换）

• 合式公式ω_1可以根据合式公式 $\omega_1 \wedge \omega_2$ 推出（∧消除）

• 合式公式 $\omega_1 \vee \omega_2$ 可以根据单个合式公式ω_1或单个合式公式ω_2推出 （∨引入）

• 合式公式ω_1可以根据合式公式¬（¬ω_1）推出（¬ 消除）

13.4 验证定义

合式公式序列 $\{\omega_1, \omega_2, \cdots, \omega_n\}$ 被称为是从一个合式公式集合Δ得出的验证（*proof*）（或演绎，*deduction*），当且仅当在序列中的每个ω_i或者是在Δ中，或者可以从处于这个序列中的较前的一个（或多个）合式公式运用若干推理规则中的一条推出。假如有一个从 Δ推出ω_n的验证，就说ω_n是集合Δ的一个定理（*theorem*）。用下面的标记来表示ω_n可以从Δ得到验证：

$$\Delta \vdash \omega_n$$

验证和定理的概念是与一个所使用的特定推理规则集合相关的。如果我们用字母$\mathcal{R}$来表示推理规则集合，那么我们可以用如下符号写出这样的事实： ω_n可以用 $\mathcal{R}$ 中的推理规则从Δ中得到验证：

$$\Delta \vdash_{\mathcal{R}} \omega_n$$

例如：给定一个合式公式的集合Δ：{P，R，P　Q}，下面的序列是一个Q ∧ R 验证，它表达了上面所述的推理规则：

{P，P　Q，Q，R，Q ∧ R }

验证的概念可以基于一个偏序，也可以基于一个序列。这种偏序可以由一个树结构来表示。验证树中的每个节点都标上一个合式公式，并且必须或者与 Δ中的一个合式公式对应，或者用若干推理规则中的一条从树的父节点推出。被标记的树是一个根节点标记的验证。图13-1 是一个验证树的例子。

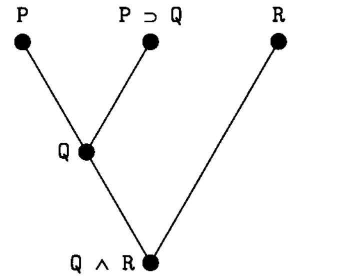

图13-1　验证树举例

13.5　语义

13.5.1　解释

现在我们来讨论一下“意义”。语义（*semantics*）必须把逻辑语言的要素与论域（*domain of discourse*）的要素联系起来。这种联系就是我们用“意义”（*meaning*）这一词所指的。就命题逻辑来说，我们把原子与关于这个世界的命题（*proposition*）联系起来（因而有命题逻辑的称谓）。所以，举例说，我们可以把原子 *BAT_OK*与命题”电池是充了电的”联系起来（我们并没有被迫要用助记原子串来作出这种联系，也可以用另外的代替）。原子与命题的联系被称为解释（*interpretation*）。在给定的解释中，与一个原子相联系的命题被称为这个原子的指称（*denotation*）。

给定一个解释，原子为真或假值。假如原子α与命题P相联系，那么对这个世界来说，只有P为真时我们才说α为真；反之它就为假。特殊原子T总是为真，而特殊原子F总是为假。既然关于这个世界的命题必须是真或者是假（在此我们愿意有这么一种理想化），我们可以通过用一种语言直接把值指派给原子来规定一个解释，而不管每个原子所指称的关于这个世界的命题。

假如一个agent有传感装置，这个装置可以用来决定各种有关这个世界的命题的真假，那么当感觉特征*x*1值为1时，与此相应的有关这个世界的命题也为真，并且与此相联系的命题逻辑原子，也许是 X1，也为真。因此，我们不用通过在某种输入“线”上的一个值 1或0来为agent表述感觉到的信息，而是能用一个在 agent的记忆结构（我们称它为知识库，*knowledge base*）中的命题演算原子来表述它。一个原子 X1在agent的知识库中出现就意味着这个 agent把与此相联系的命题在它的世界中当做真。我们很快会看到一个 agent怎样在它的知识库中使用合式公式。

13.5.2　命题真值表

在某种解释下给定原子的值，我们可以用一个真值表（*truth table*）来计算在同样解释下的任何合式公式的值。这个真值表建立了命题联结词的语义（意义）。设ω_1和ω_2 是合式公式，那么真值表规则就是：

- 假如ω_1为真并且ω_2也为真，那么（$\omega_1 \wedge \omega_2$）就为真；否则，它就为假。
- 假如ω_1为真或者 ω_2为真，或两者都为真，那么（$\omega_1 \vee \omega_2$）就为真；否则，它就为假。
- 假如ω_1为假，那么$\neg\omega_1$就为真；否则，它就为假。

- 的语义是以∨和¬来定义的。具体地说，(ω_1 ω_2)是(¬ω_1 ∨ ω_2)的替换形式和等价形式。

这些之所以被称为真值表规则是因为它们常以列表的形式来表述，如在表13-1中那样。

表13-1 命题的真值表

ω_1	ω_2	$\omega_1 \wedge \omega_2$	$\omega_1 \vee \omega_2$	$\neg \omega_1$	ω_1 ω_2
真	真	真	真	假	真
真	假	假	真	假	假
假	真	假	真	真	真
假	假	假	假	真	真

给定了组成原子的值，我们就可以用真值表来计算合式公式的值。举一个使用真值表来计算一个合式公式的例子，假设P为假，Q为假，R为真。根据这种解释，[(P Q) R] P的值是什么？按“从内到外”计算，我们首先算出P Q的值为真；然后算出(P Q) R也为真；最后，既然P为假，那么整个表达式的值一定为假。

假如一个agent使用n种特征（与命题相对应）来描述它的世界，并且这些特征是用一个相应的n个原子的集合表述在这个agent世界模型中，那么它的世界就会有2^n种不同的情形——只要这个agent能辨别，因为n个原子都可以有真值或假值，故存在着2^n种不同的情形。这个世界所能有的每一种情形都对应于一个解释。给定n个原子的值（解释），那么这个agent就可以用真值表找到任何合式公式的值。相反的过程又是怎样的呢？假设在一个合式公式的集合中已给定了这些合式公式的值，那么这些值能导出一个惟一的解释吗？这个问题很重要，因为agent常常被给定一个在诸多特征（表达式为真的合式公式）中的约束集合，那么这个agent能否用它的语言导出一个值的指派（一个解释）给原子，并且因此决定有关这个世界的命题是否为真或为假呢？也就是说，这个公式能说明这是这个世界的2^n种情形之一吗？这个答案通常是否定的。相反，在一个合式公式集合中也许会有许多给其中每个合式公式以真值的解释。为进一步探索这个主题，现在介绍模型的概念。

13.5.3 可满足性与模型

一个合式公式在一种解释下被指派为真值，那么这种解释满足这个合式公式。一种满足一个合式公式的解释被称为这个合式公式的一个模型。一种解释满足在一个合式公式集合中的每一个合式公式被称为这个合式公式集合的模型。因为一种解释给每一个原子指派一个值，所以我们可以判定一种解释是否满足任何一个原子。我们可以用真值表来判定一种解释是否满足一个包含原子的合式公式。

在举积木的机器人的例子里，我用下面的合式公式表达了一个在某些特征中的约束：

BAT_OK ∧ LIFTABLE MOVES

假如这个合式公式如我们所期望的那样值为真，那么我们必须排除所有这样的解释，在这些解释中BAT_OK和LIFTABLE为真，而MOVES为假。每个“约束合式公式”都告诉我们一些有关这个世界可能的情形，并且排除一些可能的模型（每个模型都与这个世界的一种可能情形相对应）。一般说来，描述这个世界的合式公式越多，模型就越少！没有必要为这个事实感到吃惊。我们知道有关这个世界的东西越多，那么有关这个世界可能情形的不确定性就越小。

可能没有任何解释可以满足一个合式公式（或一个合式公式的集合），在这种情况下，这个合式公式（或合式公式的集合）被说成是不一致的（*inconsistent*）或不可满足的（*unsatisfiable*）。不可满足的合式公式的例子如 F和P ∧ ¬P（没有一种解释可以使这些合式公式为真）。一个不可满足的合式公式集合的例子是{P ∨ Q，P ∨ ¬Q，¬P ∨ Q，¬P ∨ ¬Q}（用真值表可证实没有一种解释可以使所有这些合式公式为真）。

13.5.4 永真性

假如一个合式公式在它的组成原子的所有解释下都为真，那么它是永真的（因而，一个永真的合式公式是空的，它没有告诉我们任何有关这个世界可以是怎样的情形）。下面是一些永真的合式公式的例子:

- P P（这与 ¬P ∨ P 相同）
- T
- ¬（P ∧ ¬P）
- Q ∨ T
- [（P Q） P] P
- P （Q P）

用真值表来确定一个合式公式的永真性要化费大量的时间，因为这个合式公式必须要针对所有原子值的组合来计算。

13.5.5 等价

当且仅当两个合式公式的真假值在所有的解释中都是相同的，那么我们可以说它们是等价的。用符号“≡”表示等价。可以用真值表来验证下面的等价：

- 德·摩根定律：

  $\neg(\omega_1 \vee \omega_2) \equiv \neg\omega_1 \wedge \neg\omega_2$

  $\neg(\omega_1 \wedge \omega_2) \equiv \neg\omega_1 \vee \neg\omega_2$
- 换质换位定律：

  $(\omega_1 \quad \omega_2) \equiv (\neg\omega_2 \quad \neg\omega_1)$
- 假如ω_1和ω_2是等价的，那么下面的公式是永真的：

  $(\omega_1 \quad \omega_2) \wedge (\omega_2 \quad \omega_1)$

由于这个事实，$\omega_1 \equiv \omega_2$这个标记常常被用作为（$\omega_1 \quad \omega_2$）∧（$\omega_2 \quad \omega_1$）的缩写。

13.5.6 涵蕴

如果在集合Δ中，每一个合式公式都为真的所有解释下合式公式 ω值为真，那么我们说Δ逻辑涵蕴（*logically entail*）ω，并且ω从Δ中逻辑地派生（*logically follow*），ω 是Δ的一个逻辑推论（*logical consequence*）。用符号╞来指称逻辑涵蕴，并且写为Δ╞ω。下面是一些例子。

- {P}╞ P
- {P，P Q}╞ Q
- F╞ ω（ω是任何合式公式！）
- P ∧ Q ╞ P

在最后的两个例子中，把标记稍微作了简缩。当Δ为单个时，常常这么做。

逻辑涵蕴在人工智能中是很重要的，因为它提供了一种强有力的方法来说明，如果有关一个世界的命题是真的，那么另一些所关注的命题（也许是不能被感觉到的）也必须是真的。例如，假设我们感觉一些特征，把它们与公式 BAT_OK和¬ MOVES相联系，并且用公式BAT_OK ∧ LIFTABLE ⊃ MOVES来表述有关这个世界的知识。就是说，我们有三个公式，其中两个描述一个特定的世界的情景，其中另一个描述有关这个世界的一般知识。读者可根据真值表知道¬LIFTABLE由这三个公式逻辑涵蕴。既然根据这种涵蕴，¬LIFTABLE在所有这三个公式都为真的解释中为真，那么它在我们预期的解释（*intended interpretation*）（即我们把它与机器人世界相联系的）中一定为真。所以，这个作为我们预期的解释的一部分命题，即“积木是不可举的”必须是真的！

因为涵蕴对于决定有关这个世界的命题是真还是假是一个强劲的工具，所以对我们来说，研究怎样把信息表述为合式公式，并且怎样有效地产生出涵蕴的合式公式是至关重要的。可以总用真值表的来做这件事，但是我们要寻求更简便的方法。一个富有吸引力的可以替代涵蕴的是推理（*inference*）。虽然它们是根本不同的概念，但是它们可以由合理性（*soundness*）和完备性（*completeness*）的概念联结起来。

13.6 合理性和完备性

把推理与涵蕴联系起来有两个重要的定义（现在给出定理和验证的直觉含义）：

1) 假如对任意合式公式的集合Δ和合式公式ω，$\Delta \vdash \omega$ 蕴含着 $\Delta \models \omega$，那么我们说推理规则集合 是合理的（*sound*）。

2) 假如对任意合式公式的集合Δ和合式公式ω，当有$\Delta \models \omega$时，存在用推理规则集合 从Δ推出ω的验证，那么就说 是完备的（*complete*）。

对命题演算还没有给出一个完备的推理规则，下一章会讨论这个问题。

当推理规则是合理和完备的时候，可以通过寻找一个验证而不是用真值表来确定一个合式公式是否由一个合式公式的集合导出。当推理规则合理时，假如找到了一个ω从Δ导出的验证，那么我们知道ω是从Δ逻辑地导出的。当推理规则完备时，我们知道最终能够通过一个完备的搜索过程搜寻一个验证来确定ω从Δ导出（当它如此做的时候）。不管是在命题演算还是在谓词演算（将在后面研究）中，用验证的方法来替换真值表法通常节省了巨大的计算量。

要确定一个合式公式是否由一个合式公式的集合逻辑地导出或能否由此得到验证，这是一个NP难题[Cook 1971]（即其复杂性不会比原子数的指数少）。尽管如此，还是有些易处理的特殊情况，所以，了解逻辑推理的过程是重要的。

13.7 命题可满足性问题

在大多数情况下，我们都想尝试确定一个集合Δ的所有模型都是某些合式公式ω的模型。不过有时也会试图至少找出一个Δ的模型，就是说，我们也许要说明集合中的合式公式根据同样的解释每个都是可满足的。换句话说，我们要为一个包含了Δ中所有合式公式的合取公式找到一个模型。

为一个公式找一个模型的问题是一个命题可满足性问题（*propositional satisfiability*，*PSAT*）。在下一章中要说明任何公式可以被写为一些文字析取的合取。一些文字的析取被称为一个子句，

一个写成子句合取的公式叫做合取范式（*conjunctive normal form*，*CNF*）。所以它足以解决CNF公式的命题可满足性问题。许多有趣的问题，包括那些涉及约束满足、电路综合、电路诊断和规划的问题，可以编码为CNF PSAT问题（[Selman, Kautz, & Cohen 1994]）来解决。

穷尽解决CNF PSAT问题的方法就是试着系统地给公式中的原子指派真和假，然后检查每一个指派，看所有这些子句是否在这种指派下为真。假如公式中有 n个原子，那么就有 2^n个不同的指派，所以对大的n来说，这种指派过程从计算上来说是不可行的。

命题可满足性（PSAT）问题的一些特例如2SAT和3SAT问题。kSAT问题就是要为一个子句的合取找到一个模型，最长的合取式恰好包含 k个文字。2SAT问题具有多项式复杂性，3SAT则是NP完备的。因而，一般的PSAT问题是NP完备的。但是即使是所有为解决问题的已知算法，象PSAT，在最坏的情况下，也要花费指数量级的时间，许多这样的问题却只要花费多项式的期待（*expected*）时间。事实上，就许多自然概率分布而言，PSAT问题只需要多项式的平均时间[Goldberg 1979，Purdom 和Brown 1987]。

GSAT是一个非穷尽的、贪婪的、爬山型搜索过程 [Selman， Levesque 和 Mitchell 1992，Selman 和 Kautz 1993]。这个过程通过为公式中的所有原子选择一个随机的真假值的集合开始。这个真假值的集合就是一种解释。在这种解释下值为真的子句的数量可以被标识。接下来，遍历原子的列表，并当原子的值改变时，对其中每一个原子计算值为真的子句数量的增加量。改变那些给出最大增量的原子的值，并继续。当然最大增量可能会是 0或负数，但是GSAT在任何情况下都作出改变。因为这个过程可以无止境地在一个“高地”上漫游，它就应在一定次数之后中止。既然这个过程可以在局部极大值（一种满足某些但不是所有子句的解释）上中止，它就得由另一个随机的解释重新开始去搜索一个更大的极大值。 GSAT已经用来为2000个可变的、随机产生的3SAT问题找到模型，并且已被用来解决大型 N皇后问题的命题编码。 GSAT上的随机行走变化（即WALKSAT）已被用来改进它的效率 [Selman，Kautz, & Cohen 1994，Selman，Kautz, & Cohen 1996]。

13.8 另一些重要的问题

13.8.1 语言差异

命题演算是一种形式语言，它是人工 agent用来描述它的世界的。这里总是存在着一种把非形式的数学语言和英语（在这本书中用此来谈论命题演算）与形式的命题演算语言相混淆的可能性。例如，当我们说 $\{P, P \quad Q\} \vdash Q$ 时，我们用符号 $\vdash$。这个符号不是一个命题演算语言中的符号，而是一个我们用来谈论命题演算用的语言中的符号。例如，元语言（*metalinguistic*）的符号 $\vdash$ 和 $\models$ 永远不应该与符号 相混淆。

13.8.2 元定理

除了在命题演算中的定理（由推理规则链导出），我们还有关于命题演算的定理（这些常常被叫做元定理）。这里有两个重要的关于命题演算的定理。

- 演绎定理：如果 $\{\omega_1, \omega_2, \cdots, \omega_n\} \models \omega$，那么$\{\omega_1 \wedge \omega_2 \wedge \cdots \wedge \omega_n\} \quad \omega$是永真的，反之亦然。
- 反证法：如果集合Δ有一个模型，但是$\Delta \cup \{\neg \omega\}$ 没有模型（即，没有什么解释可满足所

有在这个组成的集合中的合取范式），那么 $\Delta \models \omega$。

13.8.3 结合律

二元联结词 ∧ 和 ∨ 是可结合的，即

$(\omega_1 \wedge \omega_2) \wedge \omega_3 \equiv \omega_1 \wedge (\omega_2 \wedge \omega_3)$

$(\omega_1 \vee \omega_2) \vee \omega_3 \equiv \omega_1 \vee (\omega_2 \vee \omega_3)$

因此，在这些合取范式中，我们可以去掉括号并写成

$\omega_1 \wedge \omega_2 \wedge \omega_3$

$\omega_1 \vee \omega_2 \vee \omega_3$

上面的第一个式子叫做合取式（*conjunction*），第二个式子叫做析取式（*disjunction*）。第一个式子中的每一个 ω_i 叫做合取项（*conjunct*），第二个式子中的每一个ω_i叫做析取项（*disjunct*）。

13.8.4 分配律

$\omega_1 \wedge (\omega_2 \vee \omega_3) \equiv (\omega_1 \wedge \omega_2) \vee (\omega_1 \wedge \omega_3)$

$\omega_1 \vee (\omega_2 \wedge \omega_3) \equiv (\omega_1 \vee \omega_2) \wedge (\omega_1 \vee \omega_3)$

不要被在本章中表述的某些问题表面的简明性所误导；不能恰当地掌握和记住这些概念将给以后的学习带来困难！

习题

13.1 用真值表的方法说明$\neg (P \wedge Q) \equiv (\neg P \vee \neg Q)$。

13.2 验证如果 Δ 是前后矛盾的（即它没有模型），那么 $\Delta \models \omega$，ω可以是任何合式公式。

13.3 怎样用真值表来验证假言推理是正确的？

13.4 考虑下面7个子句：

$\neg A \vee \neg B \vee \neg C$

$\neg A \vee B$

$\neg A \vee C$

$\neg B \vee C$

$\neg B \vee A$

$\neg C \vee A$

$\neg C \vee B$

为满足一个子句的集合而建立一个模型的GSAT过程有时候会在一个局部的最大值内结束。说明这里有一个对 A、B和C的真值指派，这个指派是一个由它所满足的子句数量的局部（不是全局）的最大值。

13.5 说明N皇后问题可以被表述为一个PSAT问题（提示：为$N \times N$方块图的每一个方块（k，l）引入一个原子$q_{k,l}$。假如$q_{k,l}$为真，那么在方块（k，l）上有一个皇后；假如它为假，那么该方块是空的。现在根据原子来陈述这个问题的约束。）