第15章 谓 词 演 算

15.1 动机

命题演算有一些局限性。例如，我们不能表达这样的事实：当移动木块 *B*时，说它就是ON_B_C所断言的木块*C*上的木块。在命题演算中，原子是没有内部结构的串。在关于木块的命题中，ON_A_B和ON_B_C是完全不同的。尽管给这些原子使用了助记名称（帮助我们记住它们代表什么），但也可以使用其他命题字母，如P124和Q23。

一种更有用的语言应该是既能指称在这个世界中的事物（像木块），也能指称有关这个世界的命题。我们需要一种语言，它既有对此进行命题陈述的事物的名称，又有我们要进行陈述的命题的名称。在玩积木的世界（在此之后，称之为“积木世界”）中，也许应该有像ON_B_C ¬CLEAR_C 这样的命题，其中，CLEAR_C表示木块C上是空的。要为每个木块都表达一个这样的事实将需要几个命题公式。假如我们能够用On(x, y) ¬Clear(y) 这样简单的陈述就好了，这里*x*和*y*是能够指称任何木块的变量。

在本章中，将介绍一种叫做一阶谓词演算的语言，它有这些需要的特征。谓词演算具有一些被称为对象常量（*object constant*）、关系常量（*relation constants*）和函数常量（*function constants*）的符号，以及另外我们后面要介绍的一些结构。这些语言实体（当我们讨论语义学时）将被用于指称这个世界中的事物和有关这个世界的命题。

15.2 谓词演算语言和它的句法

首先介绍一种受限的谓词演算，后面再介绍完整的谓词演算。同样，就目前来说，不要去设法对语言的结构加上意思，这样，你将与必须操作这些结构的计算机程序处在同样的地位！

组成：

- 我们有一个对象常量的无限集合，它们是字母数字组成的字符串（常常是助记的，但是只是有助于我们而不是计算机）。本书的对象常量用一个大写字母开始或者用一个数字开始。
  例如：Aa，125，13B，Q，John，EiffelTower
- 一个所有“目（*arity*）”[⊖]函数常量的无限集合。它们是字母数字组成的字符串，总是以小写字母开头，并且总是以它们的目作为上标。
  例如：$\mathtt{fatherOf}^{1}$，$\mathtt{distanceBetween}^{2}$，$\mathtt{times}^{2}$
- 一个所有目的关系常量的无限集合。这些是以大写字母开头的字母数字组成的字符串，并且以它们的目作为上标（有时称一个关系常量为谓词）。例如：$\mathtt{B17}^{3}$，$\mathtt{Parent}^{2}$，$\mathtt{Large}^{1}$，$\mathtt{Clear}^{1}$，$\mathtt{X11}^{4}$
- 命题联结词 ∨、∧、¬ 和 ，还有分隔符（、)、[、]和分隔符,。

项

⊖ arity 是一个合成词，源于如 binary (arity=2)、tertiary (arity=3)，等等中的后缀。

- 一个对象常量是一个项（*term*）。
- 一个n目的函数常量，后面跟着处于括号中、由逗号分隔的项，是一个词项。这类词项被称为函数表达式。在表示这么一个项的时候，当它的值明显可从上下文得知时，通常省略它的目上标。我们可以把对象常量当作目为0 的函数常项。例如：fatherOf（John，Bill），times（4，plus（3，6）），Sam

合式公式

- 原子：一个n目的、处于括号中由逗号分隔的n 个词项所跟随的关系常量是一个原子（也被称为原子公式，*atomic formula*）。一个0目的关系常量省略括号。另外，当目上标的值明显可从上下文得知时，常常把它省略。
  一个原子是一个合式公式。
  例如：Greaterthan（7，2），P（A，B，C，D），Q
- 命题合式公式：任何由谓词演算构成的表达式，构造方式与命题演算从别的合式公式构成一个合式公式的方式一样，它是一个合式公式，称为命题合式公式(*propositional wff*)。
  例如：[Greaterthan(7，2)∧ Lessthan(15，4)] ∨ ¬Brother (John，Sam)∨ P
  （请记住用在这些例子中的合式公式并不一定有含意！）

把命题演算中所作的扩充（具有两个合取项以上的合取式、具有两个析取项以上的析取式、子句、子句合取的集合，等等）作为谓词演算中的合式公式。目前，我们只考虑项与合式公式，后面会介绍其他的（可以允许承诺的变量）。

15.3 语义

15.3.1 世界

现在我们有了一种语言可以用来像指称命题一样指称在这个世界中的对象。我们可以如下表达：

- 这个世界可以有无限多的对象，也叫做个体（*individual*）。这些对象可以是相当具体的，如"木块A"、"Whitney先生"、"Julius Caesar"，等等；或者它们也可以是抽象的东西，如"数字7"、"π"、"所有整数的集合"，等等；它们甚至可以是虚构的或者创造的东西（对它们是否实际存在，人们可能有争议），如"美"、"圣诞老人"、"麒麟"、"诚实"，等等。只要愿意给它一个名称，并且对它说些什么，就可以把它当作我们要谈论的这个世界中的一个实际的个体。
- 个体上的函数。我们可以有数目无限的多目函数，能映射n元个体到个体。例如：可以由有一个函数映射一个人到他或她的父亲，或者可以有一个函数映射数字10和2到商数5。
- 个体上的关系。个体可以参与任意数目的关系。这些关系中每一个都有目。(具有1目的关系被称为一种属性（*property*)）。所以，个体也许会有像"重"、"大"、"蓝"等等这样一些性质，它们也许会参与在如"比……大"、"在……之间"等等这样的关系中。要从外延上指明一个n元（*n-ary*）关系，我们就要显式地列出所有参与这种关系的n元个体。

15.3.2 解释

对谓词演算中的一个表达式的解释就是一种指派，这种指派把对象常量映射到世界中的对

象，把*n*元函数常项映射到世界中的*n*元函数，把*n*元关系常量映射到世界中的*n*元关系。这些指派被称为它们相应的谓词演算表达式的指称（*denotation*）。受对象常量指派的对象的集合被称为这种解释的域。

给定一个原子的组成部分的一种解释，当原子的词项指称的那些个体的指称关系能成立时，这个原子就为真。假如这种关系不能成立，这个原子就为假。

非原子合式公式的真值与假值可用与命题演算中同样的真值表来决定。

例如：再来考虑积木世界。这是一个存在实体为 A、B、C和Floor (地板)[⊖] 的世界。我们最终要建立一种指派，把谓词演算的要素映射到这些世界的对象中。因为我们实际上没有办法让世界的对象本身成为指派的值，所以我们想象这个世界是一个数学结构，它（在别的事物中）包括数学对象A、B、C和Floor（尽量不要被"世界是个数学结构"这样的陈述所烦扰。不管这个世界实际是什么，都不妨碍我们的目标，即我们把它考虑为一个数学结构）。

我们也可以想像处于这些对象中的关系。例如，我们可以有关系 On（在……上）和Clear（清除）。这些关系可以被参与其中的对象的 *n*元来作出外延上的定义。例如，设想我们有一个如图 15-1所示这样的积木配置。在这个世界中，关系 On由对偶<B，A>、<A，C>和<C，Floor>给出。关系Clear由单项<B>给出。

所以，在这个例子中，个体A、B、C和Floor，关系On和Clear构成了这个积木世界。为了用谓词演算来描述这个世界，使用了对象常量A、B、C和Fl，二元关系常量On，和一元关系常量Clear。为方便起见，在这里用有助于记忆的符号；也许用关系常量G0045、G123、……等等来强调用在谓词演算中的符号表达没有事先指定的意义是比较好的方法。

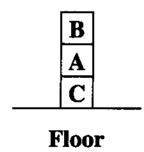

图15-1 一种积木的配置

表15-1 谓词演算与世界之间的映射

谓 词 演 算	世 界
A	A
B	B
C	C
Fl	Floor
On	On = {<B，A>，<A，C>，<C，Floor>}
Clear	Clear = {<B>}

这些谓词演算表达式要作的解释（这恰恰是许多可能解释的一种）在表 15-1中已给出。我们可以用这些指派来决定某些谓词演算合式公式的值：

On（A，B）为假，因为<A，B>不在关系On中。

Clear（B）为真，因为<B>在关系Clear中。

On（C，Fl）为真，因为<C，Floor>在关系On中。

On（C，Fl）∧ ¬On（A，B）为真，因为On（C，Fl）和 ¬ On（A，B）都为真。

15.3.3 模型及其相关的概念

有几种语义概念在谓词演算中有与在命题演算中相同的定义。让我们回顾一下：

[⊖] 当我想强调语言的元素和这些元素的指称的区别时，有时会用黑体字来表示在环境中的对象、函数及关系，而与谓词演算所用的为等宽字体。

• 如果一个合式公式在某种解释下为真，则这个解释就满足这个合式公式。
• 一个满足一个合式公式的解释就是这个合式公式的模型。
• 任何一个合式公式在所有的解释下都有真值就是永真（*valid*）。
• 任何没有模型的合式公式是不一致的或不可满足的。
• 如果一个合式公式 ω 在所有能使每一个在集合 Δ 中的合式公式都有真值的解释上为真，那么 Δ 逻辑涵蕴（*logically entails*）ω（Δ⊨ ω）。
• 当且仅当在所有的解释下两个合式公式都有相同值（即，当且仅当两个合式公式互相逻辑地涵蕴时），它们是等价的。

15.3.4 知识

谓词演算公式可以被用来表达一个 agent所具有的有关这个世界的知识。这种公式构成的一个集合 Δ 被称为这个agent的知识库，假如公式 ω 被包括在 Δ 中，我们（一般）说这个agent“知道”ω。（比较准确地说应该是这个agent“相信”ω）。

更严密地检查一下当我们说一个公式的集合包含有关这个世界的知识时可能意味着什么。例如，下面的公式是否包含了有关一个可能的积木世界的知识？

1) On（A，Fl） Clear（B）
2) Clear（B）∧ Clear（C） On（A，Fl）
3) Clear（B）∨ Clear（A）
4) Clear（B）
5) Clear（C）

使用助记法作为提示，可以很容易地构造出适合这些公式、因而也是它们的模型的积木世界的解释。图15-2中给出了我们所能想到的所有情形。在所有这三种情形中，模型使用了与用在表15-1中的把对象常量映射到对象的相同的映射。尽管如此，在这个世界中，关系常量和关系之间的映射在三个模型中是不同的。在第一个中，关系常量On被映射到关系On = {<B，A>，<A，Floor>，<C，Floor>}，关系常量Clear被映射到关系Clear = {<C><B>}。可以很容易地建立起与另外两个积木世界情形相对应的模型。

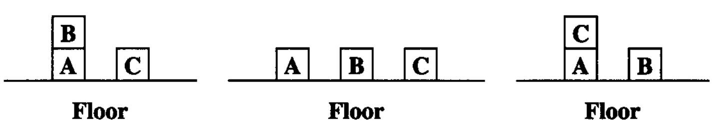

图15-2 三种积木世界情形

除了谓词演算记法提出的模型外，还可以有另外有关这些公式的模型。例如，有这样一个模型，在这个模型中，B和C是在这个世界中的同一个对象比如B的不同的谓词演算的名称。还有一些模型，在其中，A、B和C根本没有被指派给积木，而是指派给了数字⊖。这样一些可选的模型可以通过提供附加的公式来排除。例如，我们可以通过提供附加的公式 Clear（A），再假定对象常量对对象的指派已在表 15-1中给出，排除与图15-2中的两种情形相对应的模型。如在命题演算时提到的，我们拥有的公式越多，它们的模型就越少。所以，如果要约束一个公式集合的意义，以便这些公式构成有关一个特定世界的“知识”，我们必须有足够的公式使得

⊖ Löwenheim 定理指出，所有谓词演算合式公式的一致集都有一个模型，它的域是整数 [Löwenheim 1915]。

它们的模型不仅包括我们的世界，而且排除那些我们不想让它与我们的世界相混淆的世界。

15.4 量化

假定要使在域中的每一个对象有一定属性（或参与在一定的关系中）。对有限的论域，可以通过一个Clear（B1）∧ Clear（B2）∧ Clear（B3）∧ Clear（B4）这样的合取式来获得。但是长的合取式是麻烦的，而且我们不能写下无限长的合取式，虽然当论域是无限时，它也许是需要的。

或者假定在域中至少有一个对象有一定的属性。对于有限的域，这样的陈述可以通过一个析取式Clear（B1）∨ Clear（B2）∨ Clear（B3）∨ Clear（B4）来作出。然而对于大的或无限的域，问题依然存在。

为达到这样的目的，要引入新的句法实体，即变量符号（*variable symbol*）和量词符号（*quantifier symbol*）。除了先前引入的一些句法实体外，还有：

1) 我们有一个变量符号的无限集合，变量符号由靠近字母表末尾的小写字母开头的串组成，如p、q、r、s、t、…、p1、p2、…，等等（这些由它们在上下文中的使用与函数常项区分开来）。一个变量符号是一个项（除了先前定义过的）。所以，如f（x，Bob，C17）是一个三元函数表达式。

2) 我们有量词符号 和 。 称为全称量词（*universal quantifier*），而 称为存在量词（*existential quantifier*）。

3) 假如 ω 是一个合式公式而 ξ 是一个变量符号，那么（ ξ）ω 和（ ξ）ω 都是合式公式。ξ 称为被量化的（*quantified over*）的变量，而 ω 被认为是在量词的辖域内（*within the scope*）。在形如（*Q* ξ）ω 的合式公式中，ξ 是一个变量符号而*Q* 或者是 或者是 。通常，ξ 作为一个项的变量符号嵌入在ω中。假如除了ω 中的 ξ，所有的变量符号都已在 ω 中被量化的，那么（*Q*ξ）ω 被称为封闭的合式公式（*closed wff*）或封闭的语句（*closed sentence*）。在所有的应用中，合式公式是封闭的。例如：

(x) [P(x) R(x)]，(x) [P(x) (y) [R(x，y) S (f (x))]]

（在下面描述了量词语义学之后）可以说明（ *x*）[（ *y*）ω] 是与（ *y*）[（ *x*）ω]等价的，所以我们也可以把受全称量词量化的变量组成一个串放在 ω 前：（ x，y）ω。在这样的一个公式中，ω 被称为矩阵。对存在量词来说也是同样的。但是全称量词与存在量词的混合则必须保持它们的次序！（ *x*）[（ *y*）ω] 与（ *y*）[（ *x*）ω]并不等价。

也可以（在描述了下面的语义学之后）说明量词的变量是一种“哑变量”，因而可以不改变合式公式的值而被重命名。因此，假如所有出现在 ω 中的*x*都被*y*代替，（ *x*）ω 是与（ *y*）ω等价的。对存在量词来说也是如此。

既然量化变量符号给予谓词演算如此强劲的表达力，你也许会问我们是否也能量化关系符号和函数符号。在谓词演算的例子中，这样的量化是不允许的。为此，它被称为一阶谓词演算。二阶和高阶谓词演算允许量化关系和函数符号，但是要使用极其复杂的推理机制。

15.5 量词语义学

15.5.1 全称量词

在所有变量符号 ξ 对在论域中的对象的指派来说 ω（ξ）都有真值的情况下，（ ξ）ω（ ξ），

才有真值（在给定的对象常量、函数常项和关系常量对对象、函数和关系的指派下）。

例如：假设我们对Clear和On使用两个由图15-2的配置所提示的解释，在每一种这样的解释下，什么是（ x）[On（x，C） ¬Clear（C）]的真值？在这些解释中，变量*x*可以被指派给A、B、C或者Floor。我们必须为每一个解释弄清每一个这样的指派。例如，把*x*指派给A，假如On（x，C） ¬Clear（C）要有真值，我们必须让<A，C>不在关系On中，或者必须让<C>不在关系Clear中。事实上，在每一种解释中，<C>在关系Clear中，但是<A，C>不在关系On中，所以，第一个（x对A的）指派产生真值。弄清余下的指派留给你自己去做。

15.5.2 存在量词

（ ξ）ω（ξ），只是在就至少一个变量符号ξ对在论域中的对象的指派来说 ω（ξ）都有真值的情况下，才有真值（在给定的对对象、函数和关系的指派下）。

15.5.3 有用的等价式

按照约定的量词的语义学，与德·摩根定律相类似，可以建立下列等价式：

¬（ ξ）ω（ξ）≡（ ξ）¬ω（ξ）

¬（ ξ）ω（ξ）≡（ ξ）¬ω（ξ）

以及前面已经提到的：

（ ξ）ω（ξ）≡（ η）ω（η）

15.5.4 推理规则

适当地一般化后，命题演算的推理规则可以用于谓词演算。这些包括假言推理、∧的引入和消除、∨ 的引入、¬ 的消解以及归结。将在下一章介绍归结。除了命题演算规则，还有两个重要的问题：

- 全称例化（*universal instantiation*，*UI*）。从（ ξ）ω（ξ）推出ω（α），其中，ω（ξ）是具有变量 ξ 的任何合式公式，α是任何常量符号，ω（α）是 ω（ξ）用 α 在 ω 中替换 ξ 得到的。

  例如：从（ x）P（x，f（x），B）推出P（A，f（A），B）。
- 存在一般化（*existential generalization*，*EG*）。从 ω（α）推出（ ξ）ω（ξ），其中，ω（α）是一个包含常项符号 α 的合式公式，而 ω（ξ）是一个用 ξ 在 ω 中代替每一个 α 出现的形式。

  例如：从（ x）Q（A，g（A），x）推出（ y）（ x）Q（y，g（y），x）。

  容易看出UI和EG是合理的推理规则。

15.6 谓词演算作为一种表示知识的语言

15.6.1 概念化

对人工智能来说重要的是说什么，而不是怎样说它。谓词演算不过是提供一种形式统一的语言。用这种语言，有关这个世界的知识可以被表示和推断。如此表示知识的结果可以通过下一章描述的逻辑演绎方法来探索。

表示关于一个世界的知识的第一步是用对象、函数和关系把它概念化。概念化通常包含一种对概念化对象的部分进行创造的行为。常常会有种种关于我们认为什么样的对象会存在于我们的世界中的选择。我们可以自由运用任何我们所希望的方式来概念化这个世界；尽管如此，有些概念化会比另外一些更有用（未必更“精确”）。

下一步，我们构造谓词演算表达式，它要表示的意义包含对象、函数和关系。最后，我们写出在概念化了的世界中得到满足的合式公式。这些合式公式也同样可由别的解释得到满足；我们只是需要关注在某些解释下它们不会得到满足，这些解释是我们关于这个世界的知识水平所能排除的。

当设计能够推断现实世界（而不是想像的）并与其进行相互作用的 agent时，概念化要有基础是十分重要的。就是说，通过与这个世界相联系的感知机制，某些被用在知识库中的原子的真值必须是可估价的。另一些原子可以用这些初始的、感知的原子来定义，但是，假如由逻辑方法产生的结论必须与机器人存在的世界相关联，整个结构就必须依赖于某种感知。另一方面，无需用数学作基础，因为数学的论述并不一定是物质世界。

John McCarthy [McCarthy 1958]第一次正式提出在人工智能领域使用谓词演算来表示有关这个世界的知识。[Guha & Lenat 1990，Lenat 1995，Lenat & Guha 1990] 描述了一个表示数以百万计的、关于这个世界的常识的大型工程（CYC 工程)。对人工智能中的逻辑的更完整的讨论，可参考[Nilsson 1991]。而基于逻辑的人工智能的教材，可参见[Genesereth & Nilsson 1987]。

15.6.2 举例

用一些例子来说明有关一个世界的知识概念化的过程。假设我们设计了一个 agent来发送一幢大楼里的包裹。另外，我们还需要有一个关系常量，它指称某些是包裹的特性，此处用 Package（x）表示；特别地，还需要一个关系常量，它指称某一个对象在某一个房间里，用 Inroom（x，y）表示。另外还需要有一个关系常量，它指称某一个对象小于另一个对象。用这些，我们可以在谓词演算中表示下列关于这个机器人世界的例子：

- “All of the packages in room 27 are smaller than any of the packages in room 28.”

  (x，y) {[Package（x）∧ Package（y）∧ Inroom（x，27）∧ Inroom（y，28）] Smaller（x，y）}

- “Every package in room27 is smaller than one of the packages in room 29。”

  这一叙述是有歧义的（它是日常用语中的一种常见现象)。它可表示为下列两个公式的任意一个（它们不是等价的)：

  （ y）（ x）{[Package（x）∧ Package（y）∧ Inroom（x，27）∧ Inroom（y，29）] Smaller（x，y）}

  （ x）（ y）{[Package（x）∧ Package（y）∧ Inroom（x，27）∧ Inroom（y，29）] Smaller（x，y）}

一些关于机器人世界的知识也许要求我们概念化时间观念。例如，考虑这样的叙述“包裹A在包裹B前到达”。一个可推断的概念化应具有时间间隔和关于这种间隔的序列关系。我们必须有一种方法陈述一个对象的到达时间。用Arrived（x，z）表示，其中*x*指一个到达的对象，*z*指*x*到达的时间间隔。对此，我们可以写为

(z1，z2) [Arrived（A，z1）∧ Arrived（B，z2）∧ Before（ z1，z2）]

我们在第18章中更详细地讨论了一个为时间间隔的概念化过程。也有一些别的方法，有些包含已被用于计算机科学和人工智能去处理时间的时态逻辑（*temporal logic*）[Emerson 1989，Shoham 1987] 。

创造概念化会常常涉及相当棘手的问题。例如，要决定怎样处理像在陈述句“在 28号房间中的包裹包括一夸脱牛奶”中的“牛奶”这样的物质名词是很难的。牛奶是一个具有我们所说的白色这一特性的对象吗？如果是，当我们把一夸脱牛奶分成两品脱时会发生什么呢？它会变成两个对象（两个都是白的），还是仍然是一个对象？许多像这样的问题依然是留待继续研究的课题。

在本书的后面，我们要使用概念化，把无形的东西，如情形和行为，处理为对象。并将引入外延到谓词演算，使它可以允许一个agent叙述有关另一个agent的知识，如“机器人A知道包裹B在28号房间里”。

15.7 补充读物和讨论

如前所述，用逻辑语句来表示知识和用逻辑推理过程来推理已在人工智能领域引起争论，相关的讨论可参见[McDermott & Doyle 1980]。某些争议必须处理一些不匹配，它存在于归因于[Tarski 1935] （英语译本见 [Tarski 1956]）的逻辑语言的精确语义学与表述真实世界知识（与数学知识相对的）所需的更易变的和更依赖上下文的观念之间。

Tarski的语义把确定世界的个体与在语言中的对象常量联系起来。另一种观点强调索引—功能(*indexical-functional*)的表述。用Agre 和Chapman [Agre & Chapman 1990，P. 21]的话说就是：

> 尽管传统上将语义表示为一个agent头脑中的符号与这个世界中具有个性化的对象之间的对应关系，我们的理论描述这个 agent与这个世界上具有索引个性化和功能个性化的实体之间的因果关系。例如，Pengi（在[Agre 和Chapman 1987]中描述）作用的实体之一是the-bee-I-am-chasing。这个实体，就它与agent（“I”）关系的定义而言，它在索引上被赋予个性。它也在功能上被赋予个性，用一个 agent正在进行的工程之一（追一只蜜蜂）来定义。与传统的表述相对照，符号BEE-34 和BEE-35将总是指称同样的两只蜜蜂，不同的蜜蜂也许是在不同时间的the-bee-I-am-chasing。

然而，如我们在以下的章节中将看到的，研究者们已经能够应用逻辑语言（带有某些扩展）对人工智能进行表述和推理。

[Enderton 1972，Pospesel 1976]是两本论述逻辑的书——前一本更具数学性，而后一本则少点形式上的东西。从可读性来看，包括示例性的计算机程序，可参见 [Barwise 和 Etchemendy 1993]。

习题

15.1 给出一个下列域是整数的公式的模型：

On (A, Fl)　Clear (B)

Clear (B) ∧ Clear (C)　On (A, Fl)

Clear (B) ∨ Clear (A)

Clear (B)

Clear (C)

15.2 **说明合式公式 (x) On (x, A) ¬ Clear (A) 在每一种由图15-2的配置所提出的解释中有真值。**

15.3 **下列合式公式有一个明显的“积木世界”的解释：**

On (C, A)

On (A, Fl)

On (B, Fl)

Clear (C)

Clear (B)

(x) [Clear (x) ¬ (y) On (y, x)]

作出另一种满足这些合式公式的合取式的解释（对象、关系和一个映射）。

15.4 **就下列每一个谓词演算表达式，说出它是否在句法上能成为一个合法的句子（合式公式），并且，如果不能，为什么？就每一个合法的句子，说出它是否永真（必然为真）、不可满足（必然为假）或是偶然的（依赖于解释）。就每一个偶然的句子，给符号一种解释使句子为真。**

1) Won (Election, Clinton) ∧ Won (Election, Bush)

2) ¬

3) (s) [Attends (s, (CS221 ∨ CS157))]

4) (x) [Set (x) (Subset (x, x) ∨ ¬ Subset (x, x))]

5) (x) [P (x) ∨ Q (x)] ¬ (x) [(¬P (x) ∨ Q (x))]

6) (x) [IsTrue (x)]

7) (x y z) [Equal (add (x, y), add (x, y, z))]

8) P (Igor) ∧ Q (Bertha) ∧ (x) [(¬P (x) ∨ Q (x))]

9) (¬ (x)) NonExistent (x)

15.5 **一个做成尖塔形的机器人寻找两块上面都是空的积木，并把一块放在另一块的上面（假如被移动的积木可以被置于别的积木和地板上）。请写出一个一阶谓词演算表达式，它可以用于决定这里是否存在两块这样的积木。**

15.6 **一个在网格世界中的机器人可以移到一个单元中，假如这个单元是四个相邻单元之一，并且是没有障碍的。在这种情况下，我们说这个单元是可用的。同时，网格世界没有“稠密空间”（如第2章中的定义）。作出谓词演算表达式，它可以用来定义何时一个单元是可用的，并描述这个“非稠密空间”的条件。**