人工智能原理与算法

第8章 - 一阶逻辑

中国科学院自动化研究所
国科大人工智能学院

雷 震

8 一阶逻辑

一阶逻辑的引入

- 世界被赋予了许多对象，其中一些对象与另一些对象相关，需要在此基础上进行推理。
- 命题逻辑通过真值表判断蕴含性，从而实现简单的推理，然而它的表达能力有限。因此在本章中引入**一阶逻辑**，它可以简洁地表达更多东西。

- 回顾表示
- 一阶逻辑的语法和语义
- 使用一阶逻辑
- 一阶逻辑中的知识工程

- 回顾表示
- 一阶逻辑的语法和语义
- 使用一阶逻辑
- 一阶逻辑中的知识工程

8.1 回顾表示

命题逻辑的缺陷

- 命题逻辑缺乏能够简洁描述具有多个对象的环境的表达能力。
- 在Wumpus世界中表达“在与无底洞直接相邻的方格内，Agent 能感知到微风”，使用命题逻辑需要为每个格子分别写出关于微风和无底洞的规则：

$B_{1,1} \Leftrightarrow (P_{1,2} \vee P_{2,1})$（[1,1]有微风 当且仅当 [1,2]或[2,1]有无底洞）

- 4*4的Wumpus世界需要16条命题逻辑才能表达出这一规则。

8.1 回顾表示 - 思想的语言

自然语言

- 相比于命题逻辑，自然语言富有表达能力，用自然语言写作一本书，只需要偶尔转用其他语言（主要是数学和图表）。
- 命题逻辑的表示是明确的，可以看到系统的内部；人在接收自然语言后会在思想中形成内在的非文字表示，因此揭示自然语言的规则是困难的。
- fMRI等技术正在尝试对大脑进行类似的操作，例如向人展示一些词语，并利用fMRI的成像作为数据进行分类，从而探索人类对知识的表示形式。

8.1 回顾表示 - 结合形式语言和自然语言的优点

自然语言中的元素

- 自然语言中，最明显的元素是**指代对象的名词和名词性短语**（方格、无底洞、wumpus）和**表示对象之间关系的动词和动词短语**（有微风、相邻、射击）。
- 有的关系是**函数**，对于一个给定“输入”只有一个“值”。
- 对象、关系和函数的实例：
  - 对象：人、房屋、数字、理论、麦当劳叔叔、颜色、棒球游戏。
  - 关系：可以是一元关系或属性，如红色的、圆的、主要的等，或更为普适的 n 元关系，如是……的兄弟、大于、在……里、是……的一部分、有……颜色、发生于……之后、拥有、在……中间等。
  - 函数：……的父亲、……最好的朋友、……的第三局比赛、比……多一个、……的开始等。

8.1 回顾表示 - 结合形式语言和自然语言的优点

断言的表示

- 每条断言都涉及对象和属性或者关系。
  - “1+2=3”

    对象：1、2、3、1 加 2。**关系**：等于。**函数**：加。
  - “与 wumpus 相邻的方格有臭味”

    对象：wumpus、方格。**属性**：有臭味。**关系**：相邻。
  - “邪恶的约翰国王在 1200 年统治英格兰”

    对象：约翰、英格兰、1200 年。**关系**：统治。**属性**：邪恶的、国王。

8.1 回顾表示 - 结合形式语言和自然语言的优点

一阶逻辑

命题逻辑和一阶逻辑的最根本区别：每种语言所给出的本体论约定——即关于现实本质的假设不同。

- 命题逻辑假定世界中的事实要么成立要么不成立；每个事实只能处于真或假两种状态之一。
- 一阶逻辑的假设更多：世界由对象构成，对象之间的某种关系或者成立或者不成立。

语言	本体论约定（世界上存在的东西）	认识论约定（智能体对事实的认识）
命题逻辑	事实	真/假/未知
一阶逻辑	事实，对象，关系	真/假/未知
时态逻辑	事实，对象，关系，时间	真/假/未知
概率论	事实	信念度∈[0,1]
模糊逻辑	具有真实度∈[0,1] 的事实	已知的区间值

- 回顾表示
- 一阶逻辑的语法和语义
- 使用一阶逻辑
- 一阶逻辑中的知识工程

8.2 一阶逻辑的语法和语义

一阶逻辑模型示例

相比于命题逻辑，一阶逻辑的主要特征是具有对象。首先引入一个一阶逻辑模型，具有5个对象和他们之间关系的实例。

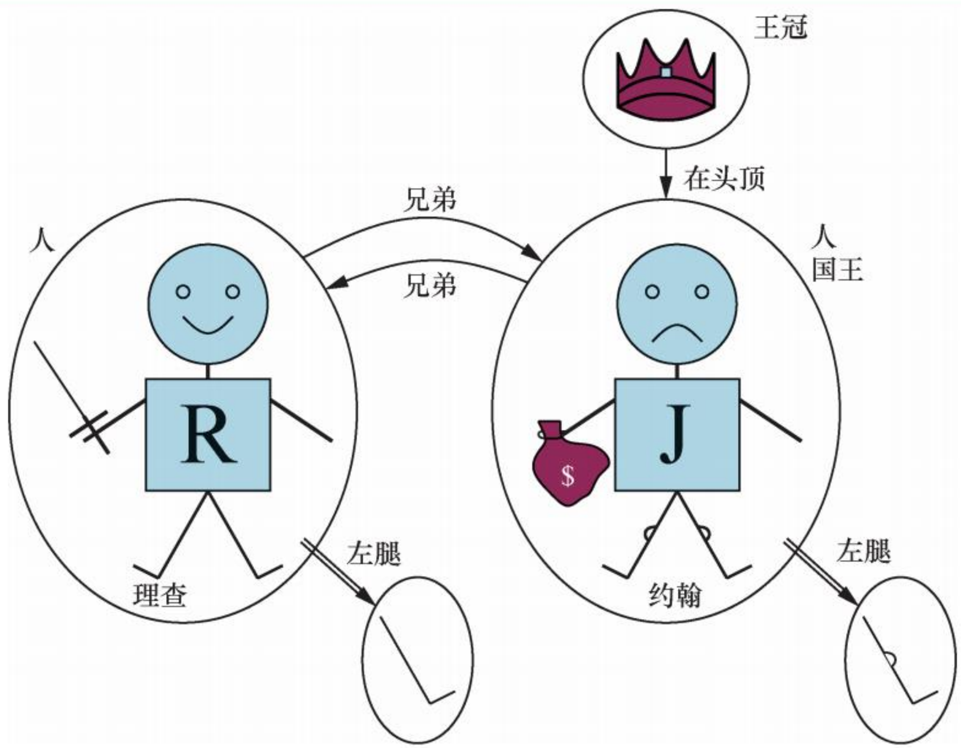


图 8-2　含有 5 个对象、2 个二元关系（兄弟和在头顶）、3 个一元关系（人、国王和王冠）和 1 个一元函数（左腿）的模型

8.2 一阶逻辑的语法和语义

一阶逻辑中的概念

- **域：**一阶逻辑模型包含的对象集或域元素的集合。
- **关系**：关系是相互关联的对象的元组集合。例如，示例模型中的兄弟关系（二元关系）就是集合

{<狮心理查, 约翰国王>,<约翰国王, 狮心理查>}

此外，模型中还含有**一元关系**，或称为属性："人"属性对于理查和约翰为真；"国王"属性仅对于约翰为真；"王冠"属性仅对王冠为真。

某些类型的关系最好视为**函数**，因为这样对于给定的对象必定仅关联到一个对象，例如，一个一元"左腿"函数，包括如下的映射：

<狮心理查>→理查的左腿

<约翰国王>→约翰的左腿

8.2 一阶逻辑的语法和语义

符号与解释

一阶逻辑的基本句法元素是表示对象、关系和函数的符号。这些符号的起始字母都大写。符号分为 3 种：

- 代表对象的**常量符号**：Richard（理查）和 John（约翰）
- 代表关系的**谓词符号**：Brother（是兄弟）、OnHead（在头顶）、Person（是人）、King（是国王）和 Crown（是王冠）
- 代表函数的**函数符号**：LeftLeg（返回输入对象的左腿）

一阶逻辑中，每个模型必须给出足够信息来确定语句的真值为真还是为假。因此，除了对象、关系和函数，每个模型还包括规范这些常量、谓词和函词的**解释**。以上括号中的内容就是一种解释。

8.2 一阶逻辑的语法和语义

指代对象的项

- **项**是指代对象的逻辑表达式。
  - 常量符号代表一个对象，因此本身就是项。
  - 对每个对象都使用不同的常量符号命名往往不太方便。例如，一般用复合项$LeftLeg(John)$来表示"约翰国王的左腿"，而不是给它取个常量符号名字。
- 项$f(t_1,t_2,...,t_n)$ 的形式语义:
  - 函词 f 指代模型中的某个函数（称为 F）
  - 参数项指代论域中的对象（称为 $d_1,d_2,...,d_n$）
  - 作为整体的项指代将函数 F 应用于 $d_1,d_2,...,d_n$ 得到的值所对应的对象。当解释确定后，每个项的指代物就确定了。

8.2 一阶逻辑的语法和语义

原子语句

有了指代对象的项以及指代关系的谓词，把它们放在一起可形成陈述事实的**原子语句**，例如：

$$Brother(Richard, John)$$

- 原子语句可以使用复合项作为参数，如：

$$Married(Father(Richard), Mother(John))$$

表明”狮心理查的父亲娶了约翰国王的母亲”。

- 如果谓词所指代的关系在参数所指代的对象中成立，那么原子语句在给定模型、给定的解释下为真。

8.2 一阶逻辑的语法和语义

复合语句

- 逻辑联结词（如否定，合取，析取）可以构建更为复杂的语句，与命题逻辑中的语法和语义一样。
- 以下4条语句，在示例给出的模型和解释下为真：

$$\neg Brother(LeftLeg(Richard), John)$$

$$Brother(Richard, John) \wedge Brother(John, Richard)$$

$$King(Richard) \vee King(John)$$

$$\neg King(Richard) \Rightarrow King(John)$$

8.2 一阶逻辑的语法和语义 - 量词

利用量词表达一般规则

- 有了允许对象存在的逻辑，那么很自然地想要表达全部对象集合的属性，而不是根据名称列举对象。
- **量词**可以帮助表达很多对象的整体属性而非根据名称逐个列举对象。一阶逻辑含有两个标准量词：
  - **全称量词** ∀：对所有对象进行陈述。
  - **存在量词** ∃：对某些对象进行陈述而不需指明其名称。

8.2 一阶逻辑的语法和语义 - 量词

全称量词

- $\forall x$：通常读作“对所有……”。“所有国王都是人”可以写作

$$\forall x\ King(x) \Rightarrow Person(x)$$

即“对于所有的x，如果x是国王，那么x是人”

- 上式中，符号x被称为**变量**，习惯上用小写字母表示。变量本身可以作为一个项，也可以作为函数的参数，例如$LeftLeg(x)$。没有变量的项被称为**基本项**。
- $\forall x\ P$表明P对每个对象x都为真，P为任意逻辑语句。更确切地说，如果P在根据一个模型的给定解释构建的所有可能扩展解释下为真，则$\forall x\ P$在该模型中为真，其中每个扩展解释给出了 x 指代的域元素。

8.2 一阶逻辑的语法和语义 - 量词

全称量词与蕴含式

- 蕴含式⇒通常用来编写含有全称量词∀的规则：断言全称量词语句等价于断言每一条蕴涵式，而最终仅对前提为真的对象断言规则的结论。
- $\forall x\ King(x) \Rightarrow Person(x)$对示例中的5个对象都下了断言，而实际只断言了约翰国王是人，对其他四个对象什么都不说。
- 一个常见的错误是使用合取式而非蕴含式与全称量词搭配，即

$$\forall x\ At(x, Berkeley)\ \wedge\ Smart(x)$$

它表示“任意x都在伯克利且是聪明的”，显然是不合理的。

8.2 一阶逻辑的语法和语义 - 量词

存在量词

- $\exists x$：读作“存在x使得......”或“对于一些x......”。$\exists x\ P$表明P至少对于一个对象x为真。准确地说，如果 P 在至少一个将x分配给域元素的扩展解释下为真，则$\exists x\ P$在给定模型中为真。
- 例如，要说约翰国王的头顶有王冠，写作

$$\exists x\ Crown(x)\ \wedge\ OnHead(x, John)$$

它相当于断言以下语句中至少有一个为真：

狮心理查是王冠 ∧ 狮心理查在约翰的头顶
约翰国王是王冠 ∧ 约翰国王在约翰的头顶
理查的左腿是王冠 ∧ 理查的左腿在约翰的头顶
约翰的左腿是王冠 ∧ 约翰的左腿在约翰的头顶
王冠是王冠 ∧ 王冠在约翰的头顶

8.2 一阶逻辑的语法和语义 - 量词

存在量词与合取式

- 正如蕴含式适合与全称量词合用一样，合取式∧是与存在量词∃合用的联结词。
- 使用蕴含式⇒与存在量词∃搭配会导致过弱的陈述，例如

$$\exists\, x\ At(x, Stanford)\ \Rightarrow\ Smart(x)$$

- 对任何不在斯坦福的人，该语句都为真（因为前提为假），因此这句话等于什么都没说。

8.2 一阶逻辑的语法和语义 - 量词

同类量词嵌套

- 使用多个量词可以表示更复杂的语句，最简单的情形是量词种类相同，连续的同类量词可以写成有多个变量的单个量词。
- 例如“兄弟是同胞”：

$$\forall x\ \forall y\ Brother(x,y) \Rightarrow Sibling(y,x)$$

也可写作$\forall x,y\ Brother(x,y) \Rightarrow Sibling(y,x)$

- 要表达同胞是对称关系，则可以写成

$$\forall x,y\ Sibling(x,y) \Leftrightarrow Sibling(y,x)$$

8.2 一阶逻辑的语法和语义 - 量词

量词混用

- 有时需要混用量词，此时量词的顺序非常重要，例如，
  - 每个人都喜爱一些人：

$$\forall x\ \exists y\ Loves(x,y)$$

  - 有人被所有人喜爱：

$$\exists y\ \forall x\ Loves(x,y)$$

- 添加括号会使语句看起来更清晰：

$$\forall x\ (\exists y\ Loves(x,y))$$

$$\exists y\ (\forall x\ Loves(x,y))$$

8.2 一阶逻辑的语法和语义 - 量词

∀与∃的联系

- 两个量词通过否定词紧密相关，二者可以转化为等价的断言。
- 断言“所有人都不喜欢胡萝卜”和“不存在喜欢胡萝卜的人”显然是等价的，即

$$\forall x\ \neg Likes(x, Parsnips) \text{ 等价于 } \neg\exists x\ Likes(x, Parsnips)$$

- 更进一步，“所有人都喜欢胡萝卜”等价于“不存在不喜欢胡萝卜的人”，即

$$\forall x\ Likes(x, Parsnips) \text{ 等价于 } \neg\exists x\ \neg Likes(x, Parsnips)$$

8.2 一阶逻辑的语法和语义 - 量词

量词的德摩根律

- 全程量词实际是对全体对象的合取，而存在量词则是析取，因此它们自然地满足德摩根律。
- 量化/非量化语句的德摩根律如下：

$$
\begin{array}{llll}
\neg\exists x\, P \equiv \forall x\, \neg P & & \neg(P \lor Q) \equiv \neg P \land \neg Q & \\
\neg\forall x\, P \equiv \exists x\, \neg P & & \neg(P \land Q) \equiv \neg P \lor \neg Q & \\
\forall x\, P & \equiv \neg\exists x\, \neg P & P \land Q & \equiv \neg(\neg P \lor \neg Q) \\
\exists x\, P & \equiv \neg\forall x\, \neg P & P \lor Q & \equiv \neg(\neg P \land \neg Q)
\end{array}
$$

- 实际上我们并不同时需要∀与∃。但可读性比简洁性重要，所以同时保留。

8.2 一阶逻辑的语法和语义 - 等词

等词符号

- 除了谓词和项，一阶逻辑还有一种构成原子语句的方式，即使用**等词符号**来表示两个项指代相同的对象：

$$Father(John) = Henry$$

表示$Father(John)$和$Henry$指代的对象相同。

- 等词也可以与否定合用，表示两项不同的对象：“理查有两个兄弟”可以写成

$$\exists x, y\ Brother(x, Richard) \wedge Brother(y, Richard) \wedge \neg(x = y)$$

$\neg(x = y)$也可用$x \neq y$简化表示。

8.2 一阶逻辑的语法和语义 - 数据库语义

逻辑语句的繁琐表述

- 承接上页的例子，假设理查有两个兄弟叫约翰和杰弗里，可写作

  $Brother(John, Richard) \land Brother(Geffery, Richard)$

- 以上语句无法完全准确地描述所需的语义：
  - 首先需要排除$John = Geffery$的情形；
  - 其次需要表达“理查有且只有两个兄弟”的语义。完整的表达：

$$Brother(John, Richard) \land Brother(Geffery, Richard) \land John \neq Geffery \land (\forall x\ Brother(x, Richard) \Longrightarrow (x = John \lor Geffery))$$

- 该语句比对应的自然语言表述烦琐很多，需要构思一种语义，使逻辑语句更加直白。

8.2 一阶逻辑的语法和语义 - 数据库语义

数据库语义

一种在数据库系统中非常流行的做法引入两个假设:

- **唯一名称假设**：每个常量符号都指代一个唯一的对象。
- **封闭世界假设**：未知其为真的原子语句事实上都为假。
- 最后，调用**域闭包**，意味着每个模型中的域元素不多于常量符号指代的那些。

由此产生的语义中，$Brother(John, Richard) \wedge Brother(Geffery, Richard)$确实能表示“理查有且仅有两个兄弟叫约翰和杰弗里”。

这种语义称为**数据库语义**，以区别于标准的一阶逻辑语义。

- 回顾表示
- 一阶逻辑的语法和语义
- 使用一阶逻辑
- 一阶逻辑中的知识工程

8.3 使用一阶逻辑 - 断言与查询

一阶逻辑的断言和查询

- 通过TELL将语句添加到知识库中，与在命题逻辑中完全一样。
- 这些语句被称为**断言**。例如，向知识库中添加约翰是国王、理查是人以及所有的国王都是人：

$$\text{TELL}(KB, King(John))$$

$$\text{TELL}(KB, Person(Richard))$$

$$\text{TELL}(KB, \forall x\ King(x) \Rightarrow Person(x))$$

8.3 使用一阶逻辑 - 断言与查询

一阶逻辑的断言和查询

- 用ASK对知识库进行提问，如

$$ASK(KB, King(John)) \text{ 返回} True$$

- 使用ASK提出的问题被称为**查询**或**目标**。知识库中逻辑蕴含的所有查询都应该得到肯定的回答：

$$ASK(KB, Person(John))$$

也应该返回True，因为它可以由$King(John)$和$\forall x\ King(x) \Rightarrow Person(x)$两条断言推断出来。

- 还可以提出量化的问题：

$$ASK(KB, \exists x\ Person(x)) \text{ 返回值为} True$$

8.3 使用一阶逻辑 - 断言与查询

ASKVARS

- 有时候True/False的返回值不是想要的回答，而是更希望数据库返回有哪些对象满足要求（如哪些对象是人，$Person(x)$为真），此时需要另外一个函数ASKVARS：

$$\text{ASKVARS}(KB, Person(x))$$

- 在本例中该函数会返回两个答案：$\{x/John\}$和$\{x/Richard\}$。这种回答叫作**置换**或者**绑定表**。

8.3 使用一阶逻辑 - 论域

论域

- 本节在一些简单的**论域**（domain）中给出范例语句。
- 在知识表示中，论域是指要表示其知识的那部分世界。
- 用三个范例论域来展示论域的构成：
  - 亲属关系
  - 数、集合、列表
  - wumpus 世界

8.3 使用一阶逻辑 - 亲属关系论域

亲属关系论域

- 亲属关系论域的对象是人，其中包含
  - 两个一元关系谓词Male和Female
  - 一系列的二元关系谓词，如：Parent、Sibling、Brother、Sister、Child、Daughter、Son、Spouse、Wife、Husband、Grandparent、Grandchild、Cousin、Aunt 和 Uncle。
  - Mother和Father用函数表示，因为从生物学角度来说，每个人只有一对父母。

8.3 使用一阶逻辑 - 亲属关系论域

亲属关系论域中的公理

- 考察每个函数和谓词，可以写出一些符号之间的关系：
  - 一个人的母亲就是他父母中的女性成员

    $$\forall m, c\ Mother(c) = m \Leftrightarrow Female(m) \wedge Parent(m, c)$$

  - 一个人的丈夫是她的男性配偶

    $$\forall w, h\ Husband(h, w) \Leftrightarrow Male(h) \wedge Spouse(h, w)$$

  - 兄弟姐妹是一个人父母的其他孩子

    $$\forall x, y\ Sibling(x, y) \Leftrightarrow x \neq y \wedge \exists p\ (Parent(p, x) \wedge Parent(p, y))$$

- 可以写出很多这样的语句，它们可以看作亲属关系论域中的**公理。** 以上这些公理同时也是定义，一般具有$\forall x, y\ P(x, y) \Leftrightarrow ...$的形式，即其他谓词对$Mother, Husband, Parent$进行了定义。

8.3 使用一阶逻辑 - 亲属关系论域

公理和定义

- 并非所有公理都是定义。
- 一些谓词没有完整的定义，因为不具有完全刻画它们的知识，例如，很难通过亲属关系来定义“人”，即很难给出语句

$$\forall x\ Person(x) \Leftrightarrow \ldots$$

- 另一方面，公理也可以是“直白的事实”，例如 $Male(Jim)$ 和 $Spouse(Jim, Laura)$。这些来自特定问题实例描述的事实使特定的提问能够得到解答。

8.3 使用一阶逻辑 - 亲属关系论域

亲属关系论域中的定理

- 并非所有关于论域的逻辑语句都是公理，其中一些是定理，它们被公理所蕴含。例如，关于兄弟姐妹关系对称性的断言：

$$\forall x, y\ Sibling(x, y) \Leftrightarrow Sibling(y, x)$$

它是可以由之前的兄弟姐妹公理推导出来的定理。

- 从纯逻辑的观点来看，知识库只需包括公理，无需包括定理，因为定理并不增加根据知识库得出的结论集；从实用观点来看，定理可以降低生成新语句的计算成本。

8.3 使用一阶逻辑 - 数、集合与列表

皮亚诺公理

- 数可能是展示从一小部分核心公理构建庞大理论的最生动的示例。这里以自然数或称非负整数的理论为例。
- 用谓词$NatNum$表示是否为自然数。此外定义常量符号0以及后继函数符号S（返回值为输入值的下一个自然数）。
- 皮亚诺公理定义了自然数和加法，其中自然数是递归定义的：

$$NatNum(0)$$

$$\forall n\ NatNum(n) \Rightarrow NatNum(S(n))$$

- 还需要加上对后继函数的约束：

$$\forall n\ 0 \neq S(n)$$

$$\forall m, n\ m \neq n \Rightarrow S(m) \neq S(n)$$

8.3 使用一阶逻辑 - 数、集合与列表

加法

- 基于后继函数，可以定义加法：

$$\forall m\ NatNum(m) \Rightarrow +(0, m) = m$$

$$\forall m, n\ NatNum(m) \wedge NatNum(n) \Rightarrow +(S(m), n) = S(+(m, n))$$

- **语法糖**：对标准语法的扩展或缩略，但不改变语义。
  - 前缀表示法：$+(m, n)$
  - 中缀表示法：$m + n$；后继函数$S(n)$写为$n + 1$

$$\forall m, n\ NatNum(m) \wedge NatNum(n) \Rightarrow (m + 1) + n = (m + n) + 1$$

- 有了加法，将乘法定义为重复的加法、乘方定义为连续的乘法就顺理成章了，同样可以定义整数除法和余数、质数等。整个数论就从一个常量、一个函数、一个谓词和 4 条公理开始构建起来。

8.3 使用一阶逻辑 - 数、集合与列表

集合

- 集合的论域对数学和常识推理也是非常重要的（实际上，可以用集合论来定义数论）。这个论域能够表示每个集合、用其他集合的元素或对其他集合的操作构建集合、判断元素是否是集合的成员、区分一个对象是否是集合。
- 使用集合论中的一般词汇来作为语法糖，包括：
  - 常量符号空集：{ }
  - 一元谓词Set，对集合为真
  - 二元谓词：包括$x \in s$，包含$s_1 \subseteq s_2$
  - 二元函数：交集$s_1 \cap s_2$，并集$s_1 \cup s_2$，添加元素$Add(x, s)$

8.3 使用一阶逻辑 - 数、集合与列表

可能的公理集

由上页规定的符号，给出一个可能的公理集：

- 集合只能是空集和向集合中添加元素产生的集合：

$$\forall s \;\; Set(s) \Leftrightarrow (s = \{\}) \vee (\exists x, s_2 \;\; Set(s_2) \wedge s = Add(x, s_2))$$

- 空集没有被加入的元素，即无法将空集分解为更小的集合和元素：

$$\neg \exists x, s \;\; Add(x, s) = \{\}$$

- 对集合添加已有元素没有作用：

$$\forall x, s \;\; x \in s \Leftrightarrow s = Add(x, s)$$

8.3 使用一阶逻辑 - 数、集合与列表

可能的公理集

- 集合中的成员只能是被添加到集合中的元素。用递归的形式表示它：声明 x 是 s 中的元素，当且仅当 s 等于某个将元素 y 添加到集合 s_2 后的集合，其中 y 与 x 相同，或 x 是 s_2的成员：

$$\forall x,s \;\; x \in s \Leftrightarrow \exists y,s_2 \;\; (s = Add(y,s_2)) \wedge (x = y \vee x \in s_2))$$

- 一个集合是另一个集合的子集当且仅当第一个集合的所有成员都是第二个集合的成员：

$$\forall s_1,s_2 \;\; s_1 \subseteq s_2 \Leftrightarrow (\forall x \;\; x \in s_1 \Rightarrow x \in s_2)$$

- 两个集合相等当且仅当它们互为对方的子集：

$$\forall s_1,s_2 \;\; (s_1 = s_2) \Leftrightarrow (s_1 \subseteq s_2 \wedge s_2 \subseteq s_1)$$

8.3 使用一阶逻辑 - 数、集合与列表

可能的公理集

- 一个对象在两个集合的交集中，当且仅当它同时是两个集合的成员：

$$\forall x, s_1, s_2 \quad x \in (s_1 \cap s_2) \Leftrightarrow (x \in s_1 \wedge x \in s_2)$$

- 一个对象在两个集合的并集中，当且仅当它是某个集合的成员

$$\forall x, s_1, s_2 \quad x \in (s_1 \cup s_2) \Leftrightarrow (x \in s_1 \vee x \in s_2)$$

8.3 使用一阶逻辑 - 数、集合与列表

列表

- 表与集合相似，它们的差别在于表中元素是有序的，同一个元素在表中出现不止一次。可以用 Lisp 语言的词汇表示列表：
  - Nil 是没有元素的空列表；Cons、Append、First 和 Rest是函数；Find 在列表中的作用与 Member 在集合中的作用相同；List 是仅对列表为真的谓词。
- 与集合一样，涉及列表的逻辑语句也常用到语法糖。
  - 空列表Nil是 []
  - Cons(x, Nil) 写作 [x]
  - 含有若干元素的列表，如 [A, B, C]，对应于嵌套项Cons(A, Cons(B, Cons(C, Nil)))

8.3 使用一阶逻辑 - wumpus 世界

wumpus 世界符号定义

本节中将介绍用简洁得多的一阶公理，自然、精确地刻画wumpus世界。

- **传感器**：
  - wumpus世界中，智能体接收一个含有 5 个元素的感知向量，分别表示臭气，微风，金光，撞击，wumpus嚷叫五种信息。
  - 通过一阶语句表示智能体接收信息时，必须包含感知和感知出现的时间，否则，智能体会搞不清何时接收到什么感知。
  - 在第5个时间步，有臭气、微风和金光，但是没有撞击或者嚷叫:

$$Percept([Stench, Breeze, Glitter, None, None], 5)$$

- **动作**：wumpus 世界中的动作可以用逻辑项表示：

$Turn(Right)$，$Turn(Left)$，$Forward$，$Shoot$，$Grab$，$Climb$

8.3 使用一阶逻辑 - wumpus 世界

wumpus 世界动作决策

- **动作查询**：要确定哪个动作最优，智能体程序执行查询：

$$AskVars(KB, BestAction(a, 5))$$

该查询将返回一个类似 {a/Grab} 的绑定表，表明Grab为最优动作。

- 原始感知数据蕴涵了关于当前状态的某些事实：

$$\forall t,s,g,w,c \quad Percept([s, Breeze, g, w, c], t) \Rightarrow Breeze(t)$$

$$\forall t,s,g,w,c \quad Percept([s, None, g, w, c], t) \Rightarrow \neg Breeze(t)$$

$$\forall t,s,b,w,c \quad Percept([s, b, Glitter, w, c], t) \Rightarrow Glitter(t)$$

$$\forall t,s,b,w,c \quad Percept([s, b, None, w, c], t) \Rightarrow \neg Glitter(t)$$

- 从而简单的“反射”行为也可以用量化蕴涵式语句来实现。例如：

$$\forall t \; Glitter(t) \Rightarrow BestAction(Grab, t)$$

8.3 使用一阶逻辑 - wumpus 世界

wumpus 世界环境表示

- 有了智能体的输入输出，接下来表示环境本身。wumpus世界中所包含的对象有方格、无底洞和 wumpus。
- **方格**：可以简单地使用坐标对应的列表项来表示，如[1,2]。方格的相邻关系可以定义为

$$\forall x, y, a, b\ Adjacent([x, y], [a, b]) \Leftrightarrow (x = a \wedge (y = b - 1 \vee y = b + 1)) \vee (y = b \wedge (x = a - 1 \vee x = a + 1))$$

- **无底洞**：用一元谓词Pit定义，在包含无底洞的方格中为真。
- **wumpus**：由于只存在一个wumpus，将其定义为常量$Wumpus$。

8.3 使用一阶逻辑 - wumpus 世界

wumpus 世界环境表示

- 智能体的位置随时间变化，因此用 At(x, s, t) 来表示对象x在时间 t 位于方格 s，从而可以表示“对象在一个时刻只能在一个位置”：

$$\forall x, s_1, s_2, t \; At(x, s_1, t) \wedge At(x, s_2, t) \Rightarrow \; s_1 = s_2$$

- 给定位置，就可以用当前的感知来推断方格的属性。例如，如果智能体在 t 时间步感知到微风，则当前所处方格是有微风的：

$$\forall \mathrm{s}, t \; At(x, s, t) \wedge \mathrm{Breeze(t)} \Rightarrow \mathrm{Breezy(s)}$$

8.3 使用一阶逻辑 - wumpus 世界

wumpus 世界环境推导

- 在能够对方格的性质进行推断后，就能进一步对无底洞、Wumpus 等的位置进行推导。例如，对于无底洞规则，可以由以下公理定义：

  $$\forall s\ Breezy(s) \land \exists r\ Adjacent(r,s) \land Pit(r)$$

  而不需要像命题逻辑那样对每个方格赋予一条公理。

- 在一阶逻辑中可以对时间进行量化：通过给谓词赋予一个后继状态公理，避免保存所有时间步的副本。对于箭的公理，可以表示为

  $$\forall t\ HaveArrow(t+1) \Leftrightarrow (HaveArrow(t) \land \neg Action(Shoot,t))$$

- 回顾表示
- 一阶逻辑的语法和语义
- 使用一阶逻辑
- 一阶逻辑中的知识工程

8.4 一阶逻辑中的知识工程

知识工程的过程

- **知识工程**：知识库构建的一般过程。
- 知识工程项目的内容、范围和难度各不相同，但所有这样的项目都包括如下的步骤：
  - 确定问题
  - 收集相关知识
  - 确定谓词、函数和常量的词汇表
  - 对论域的通用知识编码
  - 对问题实例的描述编码
  - 向推断过程提出查询并获得答案
  - 调试并评估知识库

8.4 一阶逻辑中的知识工程 - 电子电路论域

电子电路论域的构建

本节将构建一个本体论和一个知识库，进行关于下图所示的数字电路的推理。构建的过程遵循知识工程的 7 个步骤。

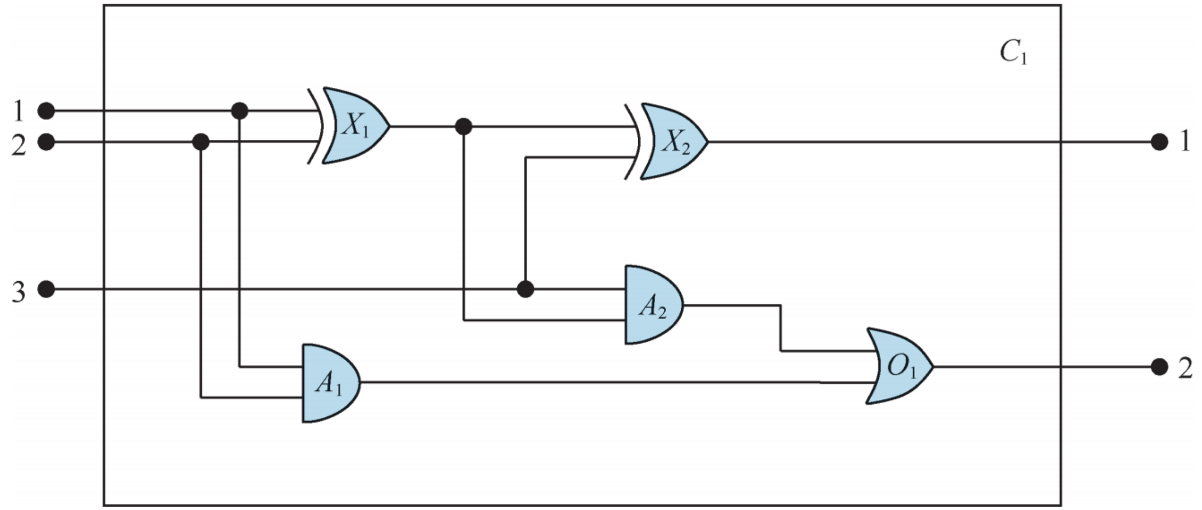


图 8-6 一位全加器的数字电路 C_1。前两个输入是要相加的两位，第三个输入是进位位。第一个输出是和，第二个输出是通往下一个加法器的进位位。电路包含两个异或门、两个与门和一个或门

8.4 一阶逻辑中的知识工程 - 电子电路论域

确定问题

- 最高层次的任务是**分析电路的功能性**：电路是否能够正确地做加法？如果所有输入都为高，A2 门的输出是什么？
- **关于电路结构的问题**：连接到第一个输入端子的门有哪些？电路是否含有反馈回路？
- 还有更详细的分析层次，包括关于延迟、电路面积、功耗以及生产成本等的分析。

8.4 一阶逻辑中的知识工程 - 电子电路论域

收集相关知识

- 获得电子电路相关的知识，包括导线和门：信号由导线传递并由门进行转换；门有四种，与门、或门、非门、异或门，不同的门以不同方式变换它的输入信号。
- 对电路的功能性进行推理将不涉及导线本身，因为重要的只是端子之间的连接关系，而不需要了解实际的连接方式。
- 目的不同，本体论也完全不同。研究故障修复需要把导线纳入本体论；解决时序故障需要纳入门延迟；考虑盈利就需要纳入价格。

8.4 一阶逻辑中的知识工程 - 电子电路论域

确定词汇表

- 这里的讨论涉及电路、端子、信号和门。选择用于表示它们的函数、谓词和常量：
  - **对门和电路进行定义**：用谓词$Gate(X)$断言X是门，用函数$Type(X)$表示门的类型，如$Type(X) = XOR$；类似地，用$Circuit(C)$断言C是电路。
  - **对端子进行定义**：用谓词$Terminal(x)$断言x是端子；$Arity(c, i, j)$断言电路c有 i 个输入端子和 j 个输出端子；用函数$In(1, X)$表示电路X的第一个输入端子，$Out(n, c)$则表示输出端子；门之间的连接性可以用谓词 Connected 表示，如$Connected(In(1, X_1), In(1, X_2))$。
  - **对信号通断进行定义**：引入两个信号值 1 和 0 作为对象，分别表示“通”和“断”，而用函数$Signal(t)$ 表示端子 t 的信号值。

8.4 一阶逻辑中的知识工程 - 电子电路论域

对论域的通用知识编码

好的本体论的标志是，只需说明少量的通用规则，就能得到清晰简洁的知识表示。需要的所有公理如下所示：

- 如果两个端子连通，则它们信号相同：

$$\forall t_1, t_2 \; Terminal(t_1) \wedge Terminal(t_2) \wedge Connected(t_1, t_2) \Rightarrow Signal(t_1) = Signal(t_2)$$

- 每个端子的信号只能是 1 或 0：

$$\forall t \; Terminal(t) \Rightarrow Signal(t) = 1 \vee Signal(t) = 0$$

- Connected 具有交换性：

$$\forall t_1, t_2 \; Connected(t_1, t_2) \Leftrightarrow Connected(t_2, t_1)$$

8.4 一阶逻辑中的知识工程 - 电子电路论域

对论域的通用知识编码

- 门的类型有 4 种：

$$\forall g\ \ Gate(g) \wedge k = Type(g) \Rightarrow k = AND \vee k = OR \vee k = XOR \vee k = NOT$$

- 与门的输出为 0，当且仅当其任意输入为 0：

$$\forall g\ \ Gate(g) \wedge Type(g) = AND \Rightarrow$$
$$Signal(Out(1, g)) = 0 \Leftrightarrow \exists n\ \ Signal(In(n, g)) = 0$$

- 或门的输出为 1，当且仅当其任意输入为 1：

$$\forall g\ \ Gate(g) \wedge Type(g) = OR \Rightarrow$$
$$Signal(Out(1, g)) = 1 \Leftrightarrow \exists n\ \ Signal(In(n, g)) = 1$$

- 异或门的输出为 1，当且仅当其输入不相同：

$$\forall g\ \ Gate(g) \wedge Type(g) = XOR \Rightarrow$$
$$Signal(Out(1, g)) = 1 \Leftrightarrow Signal(In(1, g)) \neq Signal(In(2, g))$$

8.4 一阶逻辑中的知识工程 - 电子电路论域

对论域的通用知识编码

- 非门的输出与其输入不同：

$$\forall g\ \ Gate(g) \wedge Type(g) = NOT \Rightarrow$$
$$Signal(Out(1,g)) \neq Signal(In(1,g))$$

- 除了非门之外的所有门都有两个输入和一个输出：

$$\forall g\ \ Gate(g) \wedge Type(g) = NOT \Rightarrow Arity(g,1,1)$$
$$\forall g\ \ Gate(g) \wedge k = Type(g) \wedge (k = AND \vee k = OR \vee k = XOR) \Rightarrow Arity(g,2,1)$$

- 电路有端子，数量不超过其输入和输出元数，不存在超出元数的任何东西：

$$\forall c,i,j\ \ Circuit(c) \wedge Arity(c,i,j) \Rightarrow$$
$$\forall n\ \ (n \leqslant i \Rightarrow Terminal(In(n,c))) \wedge (n > i \Rightarrow In(n,c) = Nothing) \wedge$$
$$\forall n\ \ (n \leqslant j \Rightarrow Terminal(Out(n,c))) \wedge (n > j \Rightarrow Out(n,c) = Nothing)$$

8.4 一阶逻辑中的知识工程 - 电子电路论域

对论域的通用知识编码

- 门、端子和信号是不同的：

$$\forall g,t,s \;\; Gate(g) \wedge Terminal(t) \wedge Signal(s) \Rightarrow g \neq t \wedge g \neq s \wedge t \neq s$$

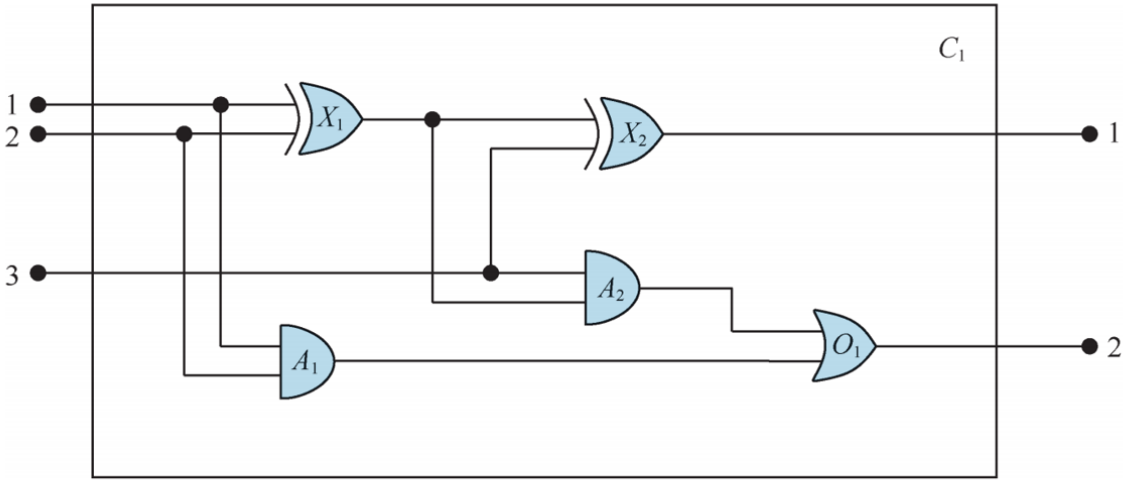
- 门是电路：

$$\forall g \;\; Gate(g) \Rightarrow Circuit(g)$$

8.4 一阶逻辑中的知识工程 - 电子电路论域

对特定问题实例编码

- 对示例中的电路C_1进行描述，该描述包括电路元件的分类，以及端子之间的连接情况：



$$Circuit(C_1) \land Arity(C_1, 3, 2)$$

$$Gate(X_1) \land Type(X_1) = XOR$$

$$Gate(X_2) \land Type(X_2) = XOR$$

$$Gate(A_1) \land Type(A_1) = AND$$

$$Gate(A_2) \land Type(A_2) = AND$$

$$Gate(O_1) \land Type(O_1) = OR$$

$$Connected(Out(1, X_1), In(1, X_2)) \qquad Connected(In(1, C_1), In(1, X_1))$$

$$Connected(Out(1, X_1), In(2, A_2)) \qquad Connected(In(1, C_1), In(1, A_1))$$

$$Connected(Out(1, A_2), In(1, O_1)) \qquad Connected(In(2, C_1), In(2, X_1))$$

$$Connected(Out(1, A_1), In(2, O_1)) \qquad Connected(In(2, C_1), In(2, A_1))$$

$$Connected(Out(1, X_2), Out(1, C_1)) \qquad Connected(In(3, C_1), In(2, X_2))$$

$$Connected(Out(1, O_1), Out(2, C_1)) \qquad Connected(In(3, C_1), In(1, A_2))$$

8.4 一阶逻辑中的知识工程 - 电子电路论域

向推断过程提出查询

- 哪种输入组合会使 C1 的第一个输出（求和位）为 0，第二个输出（进位位）为 1：

$$\exists i_1, i_2, i_3 \quad Signal(In(1,C_1)) = i_1 \wedge Signal(In(2,C_1)) = i_2 \wedge Signal(In(3,C_1)) = i_3$$
$$\wedge Signal(Out(1,C_1)) = 0 \wedge Signal(Out(2,C_1)) = 1$$

- AskVars将返回变量能使以上语句被知识库所蕴含的，变量i_1, i_2, i_3的置换，这个问题共返回3种置换：

$$\{i_1/1, i_2/1, i_3/0\} \quad \{i_1/1, i_2/0, i_3/1\} \quad \{i_1/0, i_2/1, i_3/1\}$$

- 更一般地，使用查询

$$\exists i_1, i_2, i_3, o_1, o_2 \quad Signal(In(1,C_1)) = i_1 \wedge Signal(In(2,C_1)) = i_2$$
$$\wedge Signal(In(3,C_1)) = i_3 \wedge Signal(Out(1,C_1)) = o_1 \wedge Signal(Out(2,C_1)) = o_2$$

能够返回设备的完整输入输出表。

8.4 一阶逻辑中的知识工程 - 电子电路论域

调试知识库

- 可以用很多方法来干扰知识库以便了解知识库可能的错误行为。
- 例如，假设忘记断言 $1 \neq 0$，就会发现系统除了输入 000 和 110 的情况，无法证明电路的任何输出。通过询问每个门的输出的方式可以找到问题，这里通过查询X_1的输入输出：

$$\exists i_1, i_2, o \; Signal(In(1, C_1)) = i_1 \wedge Signal(In(2, C_1)) = i_2 \wedge Signal(Out(1, X_1)) = o$$

发现 X_1 对于输入 10 和 01 没有输出（因为无法确定输入是否相同）

因此，需要告诉知识库 $1 \neq 0$。

本章小结

- 一阶逻辑表示语言，比命题逻辑表达能力更强。
- 命题逻辑只是对事实的存在进行限定，一阶逻辑对于对象和关系的存在进行限定，具有更强的表达力。
- 一阶逻辑的语法建立在命题逻辑的基础上。它增加了项来表示对象，使用全称量词和存在量词对变元进行量化来构建断言。
- 一阶逻辑的一个**可能世界**或**模型**包括一个对象集和一种将常量符号映射到对象、将谓词符号映射到对象的关系、将函数符号映射到对象上的函数的**解释**。

课程作业

假设谓词Parent(p,q)和Female(p)以及常量Joan和Kevin，字面的意思是显然的。用一阶逻辑表示下列语句。（可以用$\exists^1$表示“恰有一个”）

a. Joan有女儿（可能有多个，也可能还有儿子）。
b. Joan只有一个女儿（可能还有多个儿子）。
c. Joan只有一个孩子，是女儿。
d. Joan和Kevin只有一个儿子。

THANKS