计算机科学2004Vol. 31№. 1

互模拟与逻辑*)

沈 浩 孙永强
(上海交通大学计算机系 上海200030)

摘 要 本文首先讨论了模态逻辑与μ算子的表达能力、博弈语义,给出一阶逻辑与 Monadic 二阶逻辑的博弈形式,然后讨论互模拟等价在这些逻辑表达能力的核心作用和这些逻辑之间的关系。

关键词 互模拟,博弈,一阶逻辑,Monadic 二阶逻辑,模态逻辑,μ算子

Bisimulation with Logic

SHEN Hao SUN Yong-Qiang
(Department of Computer, Shanghai Jiao Tong University, Shanghai 200030)

Abstract This article covers expressive power of modal logic and μ-calculus, and their game semantics. We also give game semantics of first order logic and monadic second order logic, then we use bisimulation as a core concept to relate these logic.

Keywords Bisimulation, Game, FO, MSO, ML, μ

1 互模拟关系

逻辑是对系统的特性进行刻画。逻辑的表达能力可以通过它在不同模型下的区分能力来表示。首先我们要弄清楚在该逻辑下,它所谓的相同的确切意思是什么?在数学上常见的是同构等概念,而在逻辑中是通过引入在语义上适当的不变量来表示的。在一阶逻辑中是用初等等价来表示这个不变量。弄清楚在计算机科学中有用的模态逻辑与μ演算中的不变量的含义可以让我们知道这些逻辑表达能力的范围。用公式表示这种不变性,其中 M 是模型,s 是模型中的状态,L 是相应的逻辑。$(M,s)\equiv(M',s')$当且仅当对所有语句 $\varphi\in L$, $M,s\models\varphi \Leftrightarrow M',s'\models\varphi$。可以使用程序操作语义解释并刻画这种不变性进而可将此特征性质从经典逻辑中分离出来。对现实中的系统进行抽象得到的理论模型系统在计算机界使用的是传递系统。定义程序的操作语义一般有两个步骤,第一步对程序使用传递系统刻画它的运行过程,第二步根据某种观测角度定义一个等价关系,具有相同程序语义的程序在一个等价类中。程序的语义在于等价关系而不是定义它们的传递系统,我们所感兴趣的特性就是在等价系统上不可区分的特性,这要求描述这些特性的逻辑不能区分这样的等价关系。在对系统刻画时哪些特性是我们感兴趣的,我们不能列出所有感兴趣的特性,这样就要求有一个可对照的逻辑,它的表达能力经过长时间的研究可以表达所感兴趣的特性,剩下所要做的是在这个考虑的逻辑与对照逻辑之间,解决它们的表达能力关系问题。在前面提到的等价关系中,互模拟关系是重要的关系,它是通过某种观测等价定义两个系统的等价,在学习 CCS 和 π 演算时,其中最重要的概念就是互模拟的概念,并在此基础上细化为许多种互模拟关系。这个感念要追溯到逻辑学家 Johan van Benthem 在研究模态逻辑时使用的概念,这是互模拟概念第一次出现。因为传递系统展开是一颗树结构,所以我们可以在标记树上定义互模拟关系。互模拟是在两棵树节点上的一个两元关系 $R\subseteq S_{t_1}\times S_{t_2}$,满足下面条件:

- $(r_{t_1},r_{t_2})\in R$.
- 当 $(s_2,s_2)\in R$ 时我们有相同的标记 $t_1(s_1)=t_2(s_2)$。
- 如果 $s_1\rightarrow s'_1$,则存在 $s_2\rightarrow s'_2$ 并且 $(s'_1,s'_2)\in R$。
- 如果 $s_2\rightarrow s'_2$,则存在 $s_1\rightarrow s'_1$ 并且 $(s'_1,s'_2)\in R$。

两棵计算树当它们具有某个互模拟关系时就可以用互模拟等价表示 $t_1\approx t_2$。在 Kripke 模型上的互模拟关系用下面表示:$M\approx M'$ 和在状态对(s,s')上的互模拟 $(M,s)\approx(M',s')$。

2 模态逻辑与μ演算

模态逻辑公式是通过可数命题变量 $p_1,p_2,\cdots$,与命题联系词 $\neg,\wedge,\vee$,和必然操作子(necessity operator)和可能操作子(possibility operator)复合而成。记 AP 为原子命题集($|AP|\leqslant\aleph_0$)。模态逻辑的公式归纳法定义可简记如下:

$$ML::=p\mid\neg\varphi\mid\varphi\wedge\psi\mid\varphi\rightarrow\psi\mid\Diamond\varphi\mid\Box\varphi$$

其中 $p\in AP$,φ 和 ψ 为已有定义公式,◇和□是模态词。

结构框架 M 是 (A,R,F):这里的 A 是一个非空的状态集合。R 是在 A 上的二元关系,表示状态之间的转移关系。F 是函数:$p_1,p_2,\cdots\rightarrow Pow(A)$,这里的 $Pow(A)$ 是 A 的子集集合。这个模型也就是 Kripke 结构。

模态逻辑公式的语义解释,对 $s\in A$,满足关系 $M\models,\phi$,按照公式 ϕ 复杂程度定义如下:

$M\models,p_i$ 当且仅当 $s\in F(p_i)$

$M\models,\neg\phi$ 当且仅当 *not* $M\models,\phi$。

$M\models,(\phi\vee\psi)$ 当且仅当 $M\models,\phi$ 或 $M\models,\psi$。

$M\models,(\phi\wedge\psi)$ 当且仅当 $M\models,\phi$ 与 $M\models,\psi$。

$M\models,\Box\phi$ 当且仅当 $M\models,\phi$ 对所有满足 Rst 的状态成立。

$M\models,\Diamond\phi$ 当且仅当 $M\models,\phi$ 存在满足 Rst 的一个状态使之成立。

如果对所有的 $s\in A$ $M\models,\phi$ 成立,则 M 是 ϕ 的模型。模态逻辑中可能操作子与逻辑或符号都可以从别的逻辑符号导

*)基金项目:国家自然科学基金项目(60073033)。沈 浩 博士生,主要研究领域:模型检查。孙永强 教授,博导。

出。我们也可以使用 Hennessy-Milnor 形式模态词〈〉和[]。

μ 算子是在传递系统中被解释的，这里的结构框架与模态逻辑使用的是一样的。系统状态集合用 A 或 S 表示。VAR $=\{Q_1,Q_2,\cdots\}$ 是可数关系变量集合，每个关系变量 $Q\in VAR$ 可以被赋予 A 状态子集。μ 算子的公式如下：

$\mu::=p|Q|\neg f|f\wedge g|\langle a\rangle f|[a]f|\mu Q.f|\nu Q.f$

其中，$p\in AP$，f 和 g 为已有定义公式，◇和□使用 Hennessy-Milnor 形式模态词，μ 是最小不动点，ν 是最大不动点。

$p\in$原子命题集合 AP，则 p 是公式。关系变量 Q 是公式。如果 f 和 g 是公式，是$\neg f$，$f\wedge g$，$f\vee g$ 是公式。如果 f 是公式，$a\in T$，则[a]f 和〈a〉f 是公式。如果 $Q\in VAR$ 和 f 是公式，则 $\mu Q.f$ 和 $\nu Q.f$ 是公式(条件是 f 是语法单调的，就是在 f 中出现的 Q 关系变量在它之前要有偶数个$\neg$)。

公式 f 被解释为使之成立的状态集合。我们表示这样的状态集合$[f]_M$ 如下，这里 M 传递系统，$e:VAR\to 2^A$ 是赋值环境。

$e[Q\leftarrow W](Q)=W$。

$[p]_{M'}=\{s|p\in L(s)\}$。

$[Q]_{M'}=e(Q)$。

$[\neg f]_{M'}=S\backslash[f]_{M'}$。

$[f\wedge g]_{M'}=[f]_{M'}\cap[g]_{M'}$。

$[f\vee g]_{M'}=[f]_{M'}\cup[g]_{M'}$。

$[\langle a\rangle f]_{M'}=\{s|\exists\ t(s,t)\in a, a\in T \text{ and } t\in[f]_{M'}\}$。

$[[a]f]_{M'}=\{s|\forall\ t(s,t)\in a \text{ and } a\in T \text{ then } t\in[f]_{M'}\}$。

$[\mu Q.f]_{M'}$ 是谓词转换 $\tau:2^S\to 2^S$ 的最小不动点，$\tau(W)=[f]_{M'}[Q\leftarrow W]$。

$[\nu Q.f]_{M'}$ 是谓词转换 $\tau:2^S\leftarrow 2^S$ 的最大不动点，$\tau(W)=[f]_{M'}[Q\leftarrow W]$。

最大不动点与最小不动点用循环计算。

$[\mu Q.f]_{M'}=\cup_i\tau^i(\text{False}).\ \text{False}=\emptyset$。

$[\nu Q.f]_{M'}=\cap_i\tau^i(\text{True}).\ \text{True}=S$。

模态逻辑与 μ 算子的公式都有相应的否定范式(NNF)，所以只要讨论此种范式的公式就可以了。

3 逻辑中的博弈思想

前面逻辑的语义解释是用 Farski 方法，现在给出博弈解释。首先根据归纳法可证明下面的定理：

1. 如果 S 是模型 M 与 M′之间的互模拟关系并且$(s,s')\in S$ 则$(M,s)\equiv_{ML}(M',s')$。

2. 反之，若$(M,s)\equiv_{ML}(M',s')$且 M 与 M′是有限模型，则模型 M 与 M′存在互模拟关系 S 并且$(s,s')\in S$。

由互模拟关系的定义可自然看出它的博弈形式，它是在破坏者(spoiler 或 playerI)与重复者(duplicator 或 player Ⅱ)之间的游戏，可以表述成以下游戏形式。

有黑白二堆小卵石及 M 与 M′二个棋盘。不妨将二块板分别想象成 Kripke 模型 M，M′各自的外层图结构，而每个小卵石赋有一个局部的命题逻辑模型。白卵石上具有的局部命题模型种类与 Kripke 模型 M 中内层命题逻辑模型种类一样多，且对应每一种局部命题模型的卵石有充分多的拷贝以对应 M 中不同状态可能有相同标记。关于黑卵石与 Kripke 模型 M′也具有相同要求。下面是游戏操作步骤如下：

1. 破坏者先选定一种颜色，假定是白色，则从白卵石堆中取出某个卵石 W_i，将之放在板 M 的某个节点 S_i 上。如果 W_i 所赋的命题逻辑模型与 S_i 上的标记是一致的，则称 W_i 与 S_i 匹配。这步的选择是任意的，不管是色彩还是板上节点与卵石，只要匹配即可。

2. 重复者在相反一侧，就是黑色，取出相应于 W_i 的一个黑卵石 B_i，它们所赋的命题逻辑模型是一致的，放置到板 M′上的某个节点 S'_i 上。除了要求 B_i 与 S'_i 匹配以外，这步还希望满足下面的局部相似性要求：

若 $W_i(S_i)$ 是 $W_{i-1}(S_{i-1})$ 的后继，则 $B_i(S'_i)$ 也是 $B_{i-1}(S'_{i-1})$ 的后继；

若 $W_i(S_i)$ 不是 $W_{i-1}(S_{i-1})$ 的后继，则 $B_i(S'_i)$ 也不是 $B_{i-1}(S'_{i-1})$ 的后继。

如果重复者的操作能满足上述要求，则游戏继续进行下去，否者游戏结束。当重复者在相应的板上找不到相应的后继或非后继节点，或有后继但 W_i 与 B_i 不同类型，则重复者输掉比赛而破坏者赢得比赛。如果这种游戏可以一直无限地进行下去，或当破坏者在两块板上都找不到有效后继节点，而破坏者输掉比赛，从而重复者赢得比赛。如果不管破坏者怎么进行比赛，重复者都有必胜策略的话，则按必胜策略完成这个游戏后，所得的卵石对的序列或集合就是 M 与 M′之间的一个互模拟关系。反之若 M 与 M′有互模拟关系 S，则重复者可以使用 S 作为他的必胜策略。

·$M\approx M'$ 当且仅当重复者在 $G(M,M')$ 中有必胜策略。

·$(M,s)\approx(M',s')$ 当且仅当重复者在游戏 $G(M,s;M',s')$ 中有必胜策略。

μ 演算是在命题模态逻辑基础上扩展不动点算子而得到的，所以要给出 μ 演算的博弈形式最重要的是给出不动点算子的博弈形式。

根据 Tarski-Knaster 定理如果函数 $\tau:2^S\to 2^S$ 是单调函数则下面公式成立：

·$\mu Q.\tau(Q)=\cap\{Q:\tau(Q)=Q\}=\cap\{Q:\tau(Q)\subseteq Q\}$

·$\nu Q.\tau(Q)=\cup\{Q:\tau(Q)=Q\}=\cup\{Q:Q\subseteq\tau(Q)\}$

根据 Tarski-Knaster 定理可得如下等式：

·$(M,s)\models_{\nu}\nu Q.\varphi$ 当且仅当$\exists\ S\subseteq TC(s).\ s\in S$ 并且对所有 $t\in S.\ (M,t)\models_{\nu[S/Q]}\varphi$。

·$(M,s)\models_{\nu}\mu Q.\varphi$ 当且仅当$\forall\ S\subseteq TC(s)$。如果对$\forall\ t\in TC(S).\ (M,t)\models_{\nu[S/Q]}\varphi$ 可推出 $t\in S$ 则 $s\in S$。换一种说法

·$(M,s)\models_{\nu}\mu Q.\varphi$ 当且仅当$\forall\ S\subseteq TC(s)$。如果 $s\notin S$ 则$\exists\ t\in TC(s).\ t\notin S$ 并且$(M,t)\models_{\nu[S/Q]}\varphi$。

根据这样的条件可以设计出相应的对于不动点的博弈规则。其中 TC 是此状态的传递闭包，另外对应于每一个不同的不动点采用不同的颜色。

·ν：1. 选择一个状态集合。破坏者先选择一个棋盘，然后根据此不动点算子得到相应的颜色 C，把 TC(s)的一个子集涂上此颜色并且 s 也要在此集合中，然后轮到重复在另一个棋盘上采用相同的颜色，在与状态 s 认为等价的状态 s′上，把 TC(s′)中的一个子集涂上此颜色也要包含此状态 s′。

2. 选择状态。破坏者在重复者选择的状态集合中选择一个状态，而重复者在破坏者选择的状态集合中选择一个状态。

·μ：1. 选择一个状态集合。破坏者先选择一个棋盘，然后根据此不动点算子得到相应的颜色 C，把 TC(s)的一个子集涂上此颜色并且 s 不能在此集合中，然后轮到重复者在另一个棋盘上采用相同的颜色，在与状态 s 认为等价的状态 s′上，把 TC(s′)中的一个子集涂上此颜色也不能包含状态 s′。

2. 选择状态。破坏者在重复者选择的状态集合之外 TC(s)集合之内选择一个状态，而重复者在破坏者选择的状态集

合之外 TC(s′)集合之内选择一个状态。

游戏的过程构造出一个状态对序列(s_0,s_0'),…,如果保持序对的颜色匹配,则重复者赢得比赛否则是破坏者赢得比赛,其它条件类似模态逻辑上的互模拟游戏。

研究结构之间的互模拟等价用到的技巧也是博弈思想。当考虑模型的论域为有限时的模型论被称为有限模型论。模型论中许多方法在有限论域结构时不成立。Ehrenfeucht-Fraïssé 方法是幸存的方法之一。这种方法有三种形式:博弈形式、代数形式和逻辑形式(对应发明人 Ehrenfeucht,Fraïssé 和 Hintikka)。这里讨论的是博弈形式。

首先讨论一阶逻辑情况。结构 A,B 和 m∈N,公式 $A\equiv_m B$ 表示结构 A 和 B 是 m 等价(mequivalence)就是结构 A 和 B 满足相同的量词秩小于等于 m 的一阶语句。

A 和 B 的结构,让 p 是一个映射,do(p)⊆A,rg(p)⊆B,这里 do(p)是映射 p 的定义域,rg(p)是映射 p 的值域。我们说 p 是一个从 A 到 B 的部分同构(partial isomorphism),如果它满足下面条件:p 是一个一一映射。对 τ 字母表中的 c 常量:$c^A\in$do(p)和 $p(c^A)=c^B$。对 τ 字母表中的 n 维关系 R∈τ 和 $a_1,\cdots,a_n\in$do(p),下面公式成立:

$R^A a_1,\cdots,a_n$ iff R^B p$(a_1),\cdots,$p(a_n)

可以用它的图关系定义映射 p:{(a,p(a)|a∈do(p)}。用 p⊆q 表示 q 是 p 的扩展(extension)。用 Part(A,B)表示是从 A 到 B 部分同构的集合。一般来说,部分同构不能保证具有量词公式在映射后仍将成立,如果要保证具有量词的公式在映射后仍然成立就需要对它进行某种扩展。

A 和 B 都是 τ 结构,$\bar{a}\in A^r$,$\bar{b}\in B^r$ 和 m∈N。Ehrenfeucht game $G_m(A,\bar{a},B,\bar{b})$是由两个选手来玩的游戏,一个选手是破坏者(spoiler),另一个选手是重复者(duplicator)。每个选手在比赛中要轮流进行 m 轮比赛,破坏者在他的第 i 轮首先选择一个结构,然后是选择这个结构中论域中的一个成员,重复者就要选另外一个结构和其中的一个成员。这样进行 m 轮回合后,在 A 结构中有成员 $\bar{a},e_1,\cdots,e_m$,在 B 结构中有成员 $\bar{b},f_1,\cdots,f_m$,如果$\overline{ae}$到$\overline{bf}$是一个部分同构则是重复者赢,否则是破坏者赢。我们说破坏者或重复者在游戏 $G_m(A,\bar{a},B,\bar{b})$有一个必胜策略(winning strategy)是说,不管对手如何选择他都可以赢得比赛,这种策略就是必胜策略。

在二阶逻辑中最重要部分是 Monadic 二阶逻辑,其中关系变元是一维的。如果结构 A 和 B 满足相同的量词秩小于等于 m 的 Monadic 二阶句子,我们用 $A\equiv_m^{MSO}B$ 表示,就像一阶逻辑一样可以使用 Ehrenfeucht-Faïssé 博弈方法来进行刻画,MSO-G_m(A,B)。游戏规则同一阶逻辑,但现在每一步移动可以选择元素或集合。在选择元素步时,处理情况同一阶逻辑。在选择集合步时,破坏者选择子集 P⊆A 或 Q⊆B,重复者就相应选择 Q⊆B 或 P⊆A,经过 m 步后,在结构 A 中元素 $a_1,\cdots,a_r$ 和子集 $P_1,\cdots,P_t$ 与相应在结构 B 中元素 $b_1,\cdots,b_r$ 和 $Q_1,\cdots,Q_t$(m=r+s),如果重复者在游戏中可以赢得比赛,则

$\bar{a}\mapsto\bar{b}\in$Part((A,$P_1,\cdots,P_t$),(B,$Q_1,\cdots,Q_t$))

下面定理成立:

$A\equiv_m^{MSO}B$ 当且仅当重复者赢得游戏 MSO-G_m(A,B)。

4 逻辑能力的比较

前面给出模态逻辑与 μ 算子的博弈形式,然后给出经典的一阶逻辑与 Monadic 二阶逻辑的博弈形式,由于模态逻辑与 μ 算子保持某种互模拟特性,它的能力不能区分此种特性。互模拟不变性是模态逻辑与 μ 算子的关键特征性质。互模拟不变性反映了模态逻辑在识别能力上有某种模糊性,这可以从与一阶逻辑的比较可看出。恰恰是此表现识别能力的弱化使模态逻辑具有比一阶逻辑更好的语义及算法方面的优良性质。在二阶逻辑上 μ 算子也保持互模拟性质,这样 μ 算子在计算机科学上是非常有用的。Johan van Bnthm 在研究模态逻辑时使用互模拟概念,他用的是 zig-zig 关系。下面是他的重要的结论:在传递系统中互模拟不变的等价类在一阶逻辑可定义当且仅当它是模态逻辑可定义的。Igor Walukiweicz 使用自动机技巧证明在传递系统中互模拟不变的等价类在 Monadic 二阶逻辑可定义当且仅当它是 μ 算子可定义的。从以上结论可看出这几种逻辑的关系和互模拟在其中的核心作用。

参 考 文 献

1 Stirling C. Game and Modal Mu-calculus. TACAS 1996. Lecture Notes in Comuter Science 1055,1996. 298～312

2 Ebbinghaus H-D, Flum J. Finite Model Theory. Springer-Verlag, 1999

3 Milner R. Communicating and Mobile Systems: The PI-Calculus. Cambridge University Pres, March 1999

4 Shen EnShao. Set and Logic. Shanghai Jiao Tong University 2002

5 Thomas W. Language, Automata, and Logic. Handbook of Formal Language Theory (G. Rozenberg, A. Salomaa, Eds.), Vol. Ⅱ, Springer-Verlag, New York, 1999. 389～455