

基于 SMT 的不完全信息游戏求解^①



李 健, 郑荣基, 汤宇锋, 袁立然, 陈 寅

(华南师范大学 计算机学院, 广州 510631)

通讯作者: 陈 寅, E-mail: 15577316773@163.com

摘 要: 不完全信息博弈是人工智能领域的一个重要研究领域. 本文提出了一种基于可满足性模理论 (Satisfiability Modulo Theories, SMT) 的不完全信息游戏求解方法, 首先通过情景演算将游戏动态过程描述成对应的约束, 并将约束编写成命题逻辑公式, 然后将推理问题转化为逻辑公式可满足性问题, 调用 SMT 求解器 Z3 进行求解. 应用表明, 本文的算法能有效地推理出游戏的正确结果.

关键词: 可满足性模理论; 情景演算; 非完全信息博弈; Z3

引用格式: 李健, 郑荣基, 汤宇锋, 袁立然, 陈寅. 基于 SMT 的不完全信息游戏求解. 计算机系统应用, 2020, 29(1): 261-265. <http://www.c-s-a.org.cn/1003-3254/7232.html>

Solution of Incomplete Information Game Based on SMT

LI Jian, ZHENG Rong-Ji, TANG Yu-Feng, YUAN Li-Ran, CHEN Yin

(School of Computer, South China Normal University, Guangzhou 510631, China)

Abstract: Incomplete information game is an important research direction in the field of artificial intelligence. In this study, we introduce a solution of incomplete information game based on Satisfiability Modulo Theory (SMT). We describe the dynamic process of the game as corresponding constraints through Situation Calculus and write the constraints into propositional logic formula. Then, we convert the reasoning problem to the problem of satisfiability of logical formula and call the SMT solver. The application shows that the proposed algorithm can effectively deduce the answer of the game.

Key words: Satisfiability Modulo Theory (SMT); situation calculus; incomplete information game; Z3

1 概述

计算机博弈 (也称机器博弈) 是人工智能领域的重要研究方向, 是测试人工智能所达到水平的一个重要平台. 它以人工智能和各种计算机博弈技术为基础, 研究如何让计算机像人类一样从事高度智能的博弈活动. 根据掌握的信息完全程度不同, 计算机博弈可分为完全信息博弈和非完全信息博弈^[1]. 在完全信息博弈领域, 已有较为成熟的人工智能技术, 最为出名的就是国际象棋领域的“深蓝”以及围棋领域的 AlphaGo. 而对于非完全信息博弈, 由于博弈参与者掌握的信息有限, 仅

凭当前有限的信息无法准确推测对手所处的状态, 更加难以准确推断与估计对手的决策行为, 在研究上有相当大的复杂性和不确定性. 2017 年 1 月, 才由卡内基梅隆大学研发的 Libratus 人工智能系统在德州扑克中战胜人类玩家, 之后鲜有其他人工智能系统攻克非完全信息博弈.

目前, 整个非完全信息博弈的研究还处于初级阶段. 但是, 在现实生活中大到国家, 小到家庭, 人们面临的诸多问题都是以非完全信息的形式存在着. 因此, 非完全信息博弈的研究目的和意义所在, 便是如何在已

① 收稿时间: 2019-06-14; 修改时间: 2019-07-12; 采用时间: 2019-07-15; csa 在线出版时间: 2019-12-27

知的有限的信息的前提下,为人类的生活和发展提供指导,以谋求最大的利益.现在,科技高速发展,移动互联网发展更为迅猛,让人工智能和计算机博弈的发展更加的多元化.人们的碎片化时间越来越多,阅读、玩游戏等等成了人们娱乐消遣的主要方式.很多时候,玩家都是在和计算机博弈程序进行博弈^[2].针对这一种现象,很多游戏产商将游戏中的人工智能放在业务重心之一.

“谁是杀手”是一种非完全信息的推理游戏,其要求玩家根据自己手中的牌以及游戏过程中所获得的相关信息推理出杀手牌.本文以“谁是杀手”游戏为例,研究了一种基于 SMT^[3]的不完全信息游戏求解方法,旨在对非完备信息博弈进行学习与研究,同时设计并实现了一个能进行人机交互对战的网页平台.

2 背景知识

2.1 游戏简介

“谁是杀手”游戏(以下简称为“游戏”)是一种规则简单同时又需要推理的卡牌游戏.游戏共有 28 张卡牌,分为四类,每类卡牌的编号为 1 至 7. 以下我们把类型为 $X(1 \leq X \leq 4)$ 编号为 $Y(1 \leq Y \leq 7)$ 的卡牌记为 (X, Y) . 游戏可供 3-5 人玩,以下均以 3 人游戏为例. 在游戏的准备阶段,每位玩家会随机的分到 9 张牌,剩余的一张牌牌面向下放置在桌上,称为“杀手牌”. 游戏的目标是玩家猜出这张杀手牌是什么. 游戏开始后由每个玩家轮流进行行动,行动可以是以下 A0~A2 中的一种:

(1) A0 玩家说出一张牌,如果它是杀手牌,则游戏结束.

(2) A1 玩家向其它玩家请求卡牌和放置卡牌,具体来讲分为以下 3 步:

1) A1.1 玩家向其他两位玩家请求某个类型的牌.

2) A1.2 其他两位玩家如果手中有这种类型的牌,则选取一张牌面向下交给当前玩家;如果手中没有这种类型的牌,则告诉所有玩家.

3) A1.3 当前玩家根据手中的牌,如果出现 3 张不同花色但是相同序号,或者 3 张相同花色并且连续序号的手牌时,需要将其中两张暗置,只暴露其中一张来进行出牌;若手中没有符合上述情形的三张手牌时,只需出手牌中的任意一张,无需暗置.

(3) A2 玩家查阅一张.

这个游戏的一个重要的问题是,在每个回合玩家都需要判断目前的信息是否已经能够断定哪一张牌是

杀手牌.实际上,在经过若干回合后,即使能够记住每个回合的动作,由人脑来推断是否可以断定杀手牌也是很困难的.例如,在 A1.3 中如果玩家放置了 3 张牌,其中暴露的牌是 (2, 3),按照规则玩家放置的 3 张牌可以是以下 6 中组合中的任意一种:

$\{(2, 1), (2, 2), (2, 3)\}$ $\{(2, 2), (2, 3), (2, 4)\}$ $\{(2, 3), (2, 4), (2, 5)\}$ $\{(1, 3), (2, 3), (3, 3)\}$ $\{(1, 3), (3, 3), (4, 3)\}$ $\{(2, 3), (3, 3), (4, 3)\}$

例如,在 A1.3 中,如果玩家只放置了一张牌,能够用于推理的信息则更加复杂.它表示玩家手中没有任何类型的卡牌有 3 个连续的编号,同时也不存在某个编号的卡牌有 3 种类型.本文的工作就是使用逻辑工具来描述这个游戏的动态过程,并且以此为基础来进行推理.

2.2 动态系统的逻辑描述

为了进行游戏的推理,首先需要能够描述这个动态系统.情景演算 (situation calculus) 是一种用来描述动态系统的逻辑语言^[4].它在一阶逻辑的基础上增加了状态和动作,因此可以用来描述在动作的作用下不断变化的系统.情景演算中主要有 3 类元素:情景 (situation), 动作 (action) 和实体流 (fluent). 实体流用来表示不同状态下不同值的实体.

为了描述动态系统,在情景演算中包含以下的几类约束:

(1) 初始状态和约束,例如在游戏的开始,需要描述每张牌可能在某个玩家手里或者是杀手牌,以及每个玩家恰好持有 9 张牌.

(2) 描述动作的前提和效果,例如某位玩家把一张牌放置在桌上的前提是这张牌这个回合在他手上,这个动作的效果是在下个回合牌的位置在桌上.

(3) 描述持续性的约束,例如一张牌如果在前一个回合在某个玩家手上,那么如果没有相关的动作,下个回合这张牌仍然应该在这个玩家的手上.

游戏中,可以将每个玩家行动开始的时候状态作为一个情景,描述这个情景下的卡牌的位置需要满足的约束,以及玩家实际的动作所产生的约束等.为了更有效的实现系统,我们直接使用命题逻辑的公式来描述了各种约束,以便随后调用 SMT^[3]的求解器来求解.

2.3 SMT 及其求解器

可满足模理论 (Satisfiability Modulo Theory, SMT) 是判断一阶逻辑公式在组合背景理论下的可满足性问

题^[3]. SMT 的背景理论使其能很好地描述实际领域中的各种问题, 结合高效的可满足性判定算法, SMT 在测试用例自动生成、程序缺陷检测验证^[5]、RTL (Register Transfer Level) 验证^[6]、优化问题求解^[7,8]、静态分析^[9,10]、程序验证^[11,12]等一些最新研究领域中有突出的优势. SMT 求解技术的具体实现被称为 SMT 求解器 (SMT solvers). 人们研发了很多 SMT 求解器, 比较成熟的有 Z3^[13], CVC4^[14], Yices2^[15], MathSAT4^[16], Boolector^[17]. 其中, Z3 是目前为止在扩展性、表达能力以及求解效率等方面都较为出色的 SMT 求解器. 它不仅可以用来验证多个逻辑公式的可满足性, 而且可以求出一组满足约束的解. 在本文中, 我们使用 2.2 节所述方法编写命题逻辑公式, 调用 Z3 求解器来判断公式的可满足性.

3 推理的实现

假设玩家的数量为 $play_num$, 卡牌的类型为 $card_type$, 每类卡牌的数量为 $card_num$, 编号从 1 到 $card_num$. 类型为 X 编号为 Y 的卡牌记为 (X, Y) .

为了描述状态, 需要以下的谓词:

(1) $turn(T, P)$: 回合 T 的当前玩家是玩家 P .

(2) $card(T, X, Y, S)$: 卡牌 (X, Y) 在 T 回合的位置为 S , 其中,

1) $T \in N$, 表示回合数.

$$\bigwedge_{\substack{1 \leq X \leq card_type \\ 1 \leq Y \leq card_num}} \neg card(0, X, Y, 0) \wedge \bigwedge_{1 \leq P \leq player_num} AtMost(n, \{card(0, X, Y, P) \mid 1 \leq X \leq card_type, 1 \leq Y \leq card_num\}) \quad (2)$$

在每个回合的第一个阶段是由当前玩家向其他玩家请求一张线索牌. 这里有 3 种情况:

(1) 在回合 t , 玩家 p_1 从玩家 p_2 得到卡牌 (x, y) :

$$card(t-1, x, y, p_2) \wedge move(t, x, y, p_1) \quad (3)$$

(2) 在回合 t , 玩家 p_1 从玩家 p_2 得到类型 x 的卡牌, 编号未知:

$$\bigvee_{1 \leq Y \leq card_num} (card(t-1, x, Y, p_2) \wedge move(t, x, Y, p_1)) \quad (4)$$

(3) 在回合 t , 玩家 p_1 从玩家 p_2 请求类型 x 的卡牌, 但是 p_2 手中没有类型 x 的卡牌:

$$\bigvee_{\substack{1 \leq J < K \leq card_type \\ x \in \{I, J, K\}}} (((card(t-1, I, y, p) \vee move(t, I, y, p)) \wedge put(t, I, y)) \wedge ((card(t-1, J, y, p) \vee move(t, J, y, p)) \wedge put(t, J, y)) \wedge ((card(t-1, K, y, p) \vee move(t, K, y, p)) \wedge put(t, K, y))) \vee \bigvee_{\substack{1 \leq I \leq card_num-2 \\ x \in \{I, I+1, I+2\}}} (((card(t-1, x, I, p) \vee move(t, x, I, p)) \wedge put(t, x, I)) \wedge ((card(t-1, x, I+1, p) \vee move(t, x, I+1, p)) \wedge put(t, x, I+1)) \wedge ((card(t-1, x, I+2, p) \vee move(t, x, I+2, p)) \wedge put(t, x, I+2))) \quad (7)$$

2) $1 \leq P \leq play_num$, 表示玩家 P .

3) $1 \leq X \leq card_type, 1 \leq Y \leq card_num$, 表示类型为 X 编号为 Y 的卡牌.

4) $0 \leq S \leq play_num, S=0$ 表示卡牌 (X, Y) 在桌上, 否则表示卡牌在玩家 S 手中.

两个描述动作的谓词:

(3) $move(T, X, Y, P)$: 在回合 T 卡牌移动到了玩家 P 手中.

(4) $put(T, X, Y)$: 在回合 T 卡牌 (X, Y) 放到了桌上.

使用上述的描述状态和动作的谓词, 我们可以精确的描述游戏的每个回合的动作和由此产生的状态变化. 由于本游戏是一个不完全信息游戏, 因此在每个回合中, 从不同的玩家视角看到的信息是不同. 例如从玩家 1 的视角看, 如果玩家 2 请求一张类型 3 的卡牌, 那么玩家 3 交给玩家 2 的卡牌编号是未知的.

以下分别给出用来描述游戏中的动作的逻辑公式. 由于玩家视角的不同, 相同的动作由于获得的信息不同, 对应的逻辑公式可能是不同的.

游戏开始记回合数 $t=0$, 如果玩家 p 分发到的卡牌为 $(x_1, y_1), \dots, (x_n, y_n)$ 则增加公式:

$$card(0, x_1, y_1, p) \wedge \dots \wedge card(0, x_n, y_n, p) \quad (1)$$

确保所有的牌都发给了玩家, 并且每个人都拿到了 n 张卡牌:

$$\bigwedge_{1 \leq Y \leq card_num} (\neg card(t-1, x, Y, p_2)) \quad (5)$$

在每个回合的第二阶段是由当前玩家放牌在桌上. 这里有四种情况:

(1) 在回合 t , 玩家 p 把 3 张牌 $(x_1, y_1), (x_2, y_2), (x_3, y_3)$ 放在桌上:

$$\bigwedge_{i=1,2,3} ((card(t-1, x_i, y_i, p) \vee move(t, x_i, y_i, p)) \wedge put(t, x_i, y_i)) \quad (6)$$

(2) 在回合 t , 玩家 p 把 3 张牌放在桌上, 已知其中一张是 (x, y) , 其余两张未知:

(3) 在回合 t , 玩家 p 把一张牌 (x, y) 放在桌上:

$$(card(t-1, x, y, p) \vee move(t, x, y, p)) \wedge put(t, x, y) \wedge \bigwedge_{\substack{1 \leq I < J < K \leq card_type \\ 1 \leq Y \leq card_num}} (\neg(card(t-1, I, Y, p) \wedge card(t-1, J, Y, p) \wedge card(t-1, K, Y, p)))$$

$$\wedge \bigwedge_{\substack{1 \leq X \leq card_type \\ 1 \leq I \leq card_num-2}} (\neg(card(t-1, X, I, p) \wedge card(t-1, X, I+1, p) \wedge card(t-1, X, I+2, p))) \quad (8)$$

(4) 在回合 t , 玩家 p 手中无牌, 因此可以查看一张桌面上的牌, 假设是在回合 t_1 放置在桌面上的牌 (X, Y) :

$$\bigvee_{1 \leq P \leq player_num} turn(t, P) \wedge put(P, X, Y) \wedge (card(t_1 - 1, X, Y, P) \vee move(t, X, Y, P)) \quad (9)$$

如果 $A = \{a_1, \dots, a_n\}$ 是一个原子公式的集合, $k \in N$ 是一个自然数, 我们用 $AtMost(k, A)$ 表示 A 中至多有 k 个公式为真, 即如下的公式:

$$\neg \bigvee_{1 \leq i_1 < i_2 < \dots < i_k \leq n} (a_{i_1} \wedge \dots \wedge a_{i_k})$$

对于每个回合 t , 我们需要增加如下的公式来描述背景知识.

(1) 回合 t 有 1 个当前玩家

$$AtMost(1, \{turn(t, P) \mid 1 \leq P \leq player_num\}) \wedge \bigvee_{1 \leq P \leq player_num} turn(t, P) \quad (10)$$

(2) 在回合 t 至多有 n 张牌移动到来其它玩家手中

(3) 在回合 t 至多有 n 张牌放置在桌上

$$AtMost(n, \{move(t, x, y, P) \mid 1 \leq X \leq card_type,$$

$$AtMost(n, \{put(t, x, y) \mid 1 \leq X \leq card_type, 1 \leq Y \leq card_num\}) \quad (12)$$

$$1 \leq Y \leq card_num, 1 \leq P \leq player_num\}) \quad (11)$$

(4) 回合 t 结束时卡牌的位置

$$\bigwedge_{\substack{1 \leq X \leq card_type \\ 1 \leq Y \leq card_num}} (\bigvee_{0 \leq P \leq player_num} ((\neg move(t, X, Y, P) \wedge \neg put(t, X, Y, P) \wedge card(t-1, X, Y, P) \wedge card(t, X, Y, P)) \vee (move(t, X, Y, P) \wedge \neg put(t, X, Y, P) \wedge card(t, X, Y, P)) \vee ((put(t, X, Y, P) \wedge card(t, X, Y, P)))) \quad (13)$$

推理杀手牌的过程如算法 1 所示. 定义一个 SMT 求解器 Z3, 将每回合产生的约束对应的逻辑公式添加到 Z3. 在回合 0, 根据初始卡牌将式 (1) 和式 (2) 加入到 Z3. 从回合 1 开始读入数据, 并判断在每个回合是否能够确定杀手牌. 如果 Z3 求解器返回 SAT, 说明有符合约束的解, 如果将某一可能的杀手牌约束为非杀手牌后返回了 UNSAT, 说明该杀手牌是唯一可能的杀手牌, 则返回该杀手牌, 否则进行下一回合.

- 17 可能的杀手牌
- 18 $t=t+1$; // 进行下一回合

算法 1. 推理杀手牌的过程

```

1  $\Phi = \emptyset$ ;
2  $t=0$ ; // 回合 0
3 根据初始卡牌将公式 (1) 和式 (2) 加入  $\Phi$ ;
4  $t=1$ ; // 从回合 1 开始读入数据, 并判断再每个回合是否能够确定杀手牌
5 while 读取回合  $t$  的信息 do
6   根据回合  $t$  的动作, 将公式 (3)-(9) 加入  $\Phi$ ;
7   将回合  $t$  的背景知识公式 (10)-(13) 加入  $\Phi$ ;
8   if check( $\Phi$ )=UNSAT then
9     返回错误; // 读入数据有误
10  else
11     $M$  是  $\Phi$  的一个模型;
12     $(X, Y)$  是  $M$  中的杀手牌; //  $(X, Y)$  是可能的杀手牌
13  if check( $\Phi \cup \neg card(0, X, Y, 0)$ )=UNSAT then
14    return  $(X, Y)$  是杀手牌; //  $(X, Y)$  是唯一
    
```

4 系统实现

本文除了解决“谁是杀手”游戏推理问题外, 还对游戏进行了简单实现. 玩家可以登录网页进行人机对战. 系统包括客户端和服务端两部分, 系统框架如图 1 所示. 游戏玩家在客户端登陆, 与服务端的 AI 玩家进行 PK.

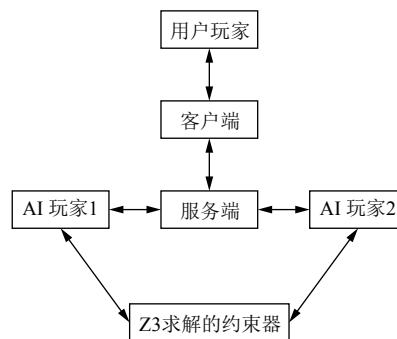


图 1 系统框架图

4.1 客户端

游戏玩家登陆网页后, 点击“准备”按钮, 客户端进

行初始化,并向服务端发送请求,服务端接收到后,模拟发牌操作,将手牌发给客户端。当客户端接收后,解析并将收到的手牌信息显示给玩家。接着玩家可以根据游戏状态点击“问牌”、“看牌”、“出牌”、“给牌”、“猜牌”按钮进行相应操作。服务端接收到客户端动作消息后模拟 AI 用户做出应答操作。

4.2 服务端

在服务端中,不仅要响应客户端的请求,还要模拟 AI 玩家游戏过程,进行“猜牌”、“问牌”、“给牌”、“出牌”、“看牌”这些操作。当轮到 AI 玩家给牌时,判断是否持有符合要求的卡牌,有则打出卡牌;当轮到 AI 玩家看牌时,随机选择某个回合放置在桌面的未知卡牌进行查看;当轮到 AI 玩家出牌时,随机打出符合要求的卡牌;当轮到 AI 玩家问牌时,随机选择一张未知卡牌。服务端会以 AI 玩家的视角,记录游戏过程。AI 玩家的推理过程如算法 1 所示。

5 总结与展望

本文提出了一种基于 SMT 的不完全信息游戏求解方法并基于此方法实现了一个“谁是杀手”游戏应用,该应用提供了一个能进行人机交互对战的网页平台。游戏用户在客户端登陆便可与服务器端的 AI 进行游戏对战。AI 玩家能够应用 Z3 求解器推理出杀手牌,但在问牌和出牌游戏动作时,没有添加任何优化策略,只是能够打出符合规则的手牌,不会去考虑游戏的利益关系,处于一种“初级”玩家的智能水平。因此,下一步研究内容是通过优化 AI 玩家问牌和出牌策略,使“出牌”的次数尽量少,同时获得的线索信息尽可能更多,从而加强 AI 玩家的智能程度,加快推理过程。

参考文献

- 代佳宁. 基于虚拟遗憾最小化算法的非完备信息机器博弈研究[硕士学位论文]. 哈尔滨: 哈尔滨工业大学, 2014.
- 李景棚. 非完备信息博弈估值算法的研究[硕士学位论文]. 哈尔滨: 哈尔滨工业大学, 2014. 1-3.
- 王翀, 吕荫润, 陈力, 等. SMT 求解技术的发展及最新应用研究综述. 计算机研究与发展, 2017, 54(7): 1405-1425. [doi: 10.7544/issn1000-1239.2017.20160303]
- Lin FZ. Chapter 16 situation calculus. Foundations of Artificial Intelligence, 2008, 3: 649-669. [doi: 10.1016/S1574-6526(07)03016-7]
- Cordeiro L, Fischer B, Marques-Silva J. SMT-based bounded model checking for embedded ANSI-C software. IEEE Transactions on Software Engineering, 2009, 38(4): 957-974.
- Kunapareddy S, Turaga SD, Sajjan SSTM. Comparison between LPSAT and SMT for RTL verification. 2015 International Conference on Circuits, Power and Computing Technologies (ICCPCT-2015). Nagercoil, India. 2015. 1-5.
- Sebastiani R, Tomasi S. Optimization in SMT with LA(Q) cost functions. International Joint Conference on Automated Reasoning. Berlin, Heidelberg. 2012. 484-498.
- 陈力, 王永吉, 吴敬征, 等. 基于树状线性规划搜索的单调速率优化设计. 软件学报, 2015, 26(12): 3223-3241. [doi: 10.13328/j.cnki.jos.004853]
- Blackham B, Liffiton M, Heiser G. Trickle: Automated infeasible path detection using all minimal unsatisfiable subsets. IEEE Real-Time and Embedded Technology and Applications Symposium. Berlin, Germany. 2015. 169-178.
- Eldib H, Wang C, Taha M, et al. Quantitative masking strength: Quantifying the power side-channel resistance of software code. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 2015, 34(10): 1558-1568. [doi: 10.1109/TCAD.2015.2424951]
- Leroy X. Formal verification of a realistic compiler. Communications of the ACM, 2009, 52(7): 107-115. [doi: 10.1145/1538788.1538814]
- Klein G. Operating system verification —An overview. Sadhana, 2009, 34(1): 27-69. [doi: 10.1007/s12046-009-0002-4]
- De Moura L, Bjørner N. Z3: An efficient SMT solver. Proceedings of International Conference on Tools and Algorithms for the Construction and Analysis of Systems. Berlin, Heidelberg, Germany. 2008. 337-340.
- Barrett C, Conway CL, Deters M, et al. CVC4 //International Conference on Computer Aided Verification. Berlin, Heidelberg, Germany. 2011. 171-177.
- Dutertre B. Yices 2.2. International Conference on Computer Aided Verification. Cham. 2014. 737-744.
- Bruttomesso R, Cimatti A, Franzén A, et al. The mathsat 4 smt solver. International Conference on Computer Aided Verification. Berlin, Heidelberg, Germany. 2008. 299-303.
- Brummayer R, Biere A. Boolector: An efficient SMT solver for bit-vectors and arrays. International Conference on Tools and Algorithms for the Construction and Analysis of Systems. Berlin, Heidelberg, Germany. 2009. 174-177.