软件学报 ISSN 1000-9825, CODEN RUXUEW
Journal of Software, [doi: 10.13328/j.cnki.jos.000000]

E-mail: jos@iscas.ac.cn
http://www.jos.org.cn
Tel: +86-10-62562563

自主移动机器人空间永恒探索算法的符号模型检测方法*

蔡晓伟[1,2], 张 民[1,2]

[1](华东师范大学 上海市高可信计算重点实验室,上海 普陀 200062)

[2](华东师范大学 教育部高可信软件国际联合实验室,上海 普陀 200062)

通讯作者: 张民, E-mail: zhangmin@sei.ecnu.edu.cn

摘 要: 随着物联网技术的发展,自主移动机器人(autonomous mobile robots)在网络中的作用也日益重要,利用形式化的方法验证自主移动机器人行为的正确性逐渐成为新的研究热点.当前主要的验证方法多以初始状态已知为前提且面临状态爆炸问题.本文以自主移动机器人空间永恒探索算法为例,提出自主移动机器人符号模型检测方法,该方法不依赖某个具体的初始状态,且适用于不同的同步模型.同时,借助符号模型检测的高效性,有效避免状态爆炸问题.利用 nuXmv 符号模型验证工具对机器人探索算法在三种同步模型:完全同步模型 FSYNC(Full-synchronous model)、半同步模型 SSYNC(semi-synchronous model)、异步模型 ASYNC(Asynchronous model)进行建模并利用 LTL 公式定义算法的性质,最终实现算法的形式化验证.验证结果表明在假设初始状态未知的条件下依然可验证性质不被满足并找到反例.同时,实验数据表明了符号模型检测对自主移动机器人算法形式化验证的可行性与高效性.

关键词: nuXmv;移动机器人;空间探索;符号模型检测.

中图法分类号: TP311

中文引用格式: 蔡晓伟,张民. 自主移动机器人空间永恒探索算法的符号模型检测方法. 第二届全国形式化方法与应用会议(FMAC 2017). http://www.jos.org.cn/1000-9825/ 0000.htm

英文引用格式: Xiaowei Tsai, Min Zhang. On Symbolic Model Checking of Mobile Robots Perpetual Exploration Algorithm. 2nd National Symposium on FMAC, 2017 (in Chinese). http://www.jos.org.cn/1000-9825/0000.htm

On Symbolic Model Checking of Mobile Robots Perpetual Exploration Algorithm

CAI Xiaowei[1,2], ZHANG Min[1,2]

[1](Shanghai Key Lab of Trustworthy Computing, ECNU, Shanghai 200062, China)

[2](MoE International Joint Lab of Trustworthy Software, ECNU, Shanghai 200062, China)

Abstract: With the development of Internet of Things technology, the role that autonomous mobile robots play in the network is becoming increasingly important. Formal verification of the correctness of autonomous mobile robots has become a new research topic. Most of the exising approaches either suffer state-explosion problem or rely on concrete initial states, which however are usually not undefined and have to be enumerated to make verification complete. In this paper, we propose a symbolic model checking approach to the formal verification of a typical automonous mobile robot system called mobile robot perpetual exploration system. For the symbolicity feature of the model checking, the verifcation does not rely on specific initial states and meanwhile state-explosion problem can be avoided. We use the state-of-the-art symbolic model checker nuXmv to verify the mobile robot perpetual exploration algorithm under three different scheduling modes called full synchronous model (FSYNC), semi-synchronous model (SSYNC) and asynchronous model (ASYNC). Experimental results show that even without providing specific initial states a counterexample can be found in our approach for the perpetual exploration property, which concides with the existing verification result which is obtained by model checking with specific

* 基金项目: 国家自然科学基金(61502171); 上海市科委重点/重大项目(No. 15511104700, No. 16DZ1100600).

Foundation item: National Natural Science Foundation of China (61502171); the Municipality Projects of Shanghai Science and Technology Commission (No. 15511104700, No. 16DZ1100600).

收稿时间: 0000-00-00; 修改时间: 0000-00-00; 采用时间: 0000-00-00; jos 在线出版时间: 0000-00-00

CNKI 在线出版时间: 0000-00-00

initial states. Meanwhile, the experimental data shows the feasibility and efficiency of symbolic model checking in the formal verification of autonomous mobile robot systems.

Key words: nuXmv; LTL; Mobile robots; space exploration; symbolic model checking

随着物联网技术的飞速发展,自主移动机器人(Automonous mobile robots)在网络中的作用日益重要.自主移动机器人可以在没有任何中央调度的情况下,通过互相协调合作完成任务[1],用于一些特殊环境,如未知区域的地图构造、环境监测、城市搜索、在人无法进入的危险区域作业、监视与未知空间探索等.

移动机器人空间探索协议(Mobile robot exploration protocol)[2-5]是一种典型的自主移动机器人协议.通过定义自主机器人的行为,实现物理空间的探索.在机器人自主空间探索协议中,机器人之间无法通过外部表征进行识别,所有机器人是完全相同的且都执行相同的探索算法.机器人具备以下几个特征[6]:1、无记忆存储,即不能存储过去完成的动作;2、无方向传感器,即本身无法辨别方向,并且没有偏好移动方向;3、无通讯功能,即机器人之间不能发送和接收消息;4、有视觉传感器,即可以通过视觉传感器获取空间其他机器人的位置信息.移动空间模型由原始的连续二维欧几里德空间模型,逐渐演化成为有限位置的离散空间模型.离散空间使用图来描述,图的结点代表空间的位置,边表示机器人可以由一个结点到相邻结点的路径.离散空间模型简化了机器人模型,更加关注机器人的数量和空间位置之间的关系.

机器人的移动可以分为三个阶段:观察(Look),计算(Compute),移动(Move).在观察阶段,机器人可以获取图的快照信息,这些快照信息记录着其他机器人在图上的位置.在收集到其他机器人位置信息之后,机器人进入计算阶段,依据收集到的信息计算决定是否移动.移动阶段完成之前计算阶段做出的移动策略.原始模型中,部分机器人同步执行观察、计算、移动,并且这个三个阶段具有原子性,这种模型下,调度策略有两种:完全同步调度模型 FSYNC (fully-synchronous model)和半同步调度模型 SSYNC (semi-synchronous model).随后,Flocchini 等人提出了异步调度模型 ASYNC (asynchronous model)[7,8].该模型中,每个机器人观察,计算,移动三个阶段不再具有原子性,每个机器人在异步执行移动阶段可能会使用过时的快照信息做出移动决策.自主机器人空间探索协议有两个变体,包括探索终止(exploration with stop)和永恒探索(perpetual exclusive exploration).探索终止即所有的机器人最终都会在某个位置停止探索;而永恒探索表示所有的机器人将不停地探索所有可能的节点.

自主机器人空间探索协议核心问题是根据具体的物理空间如何定义自主移动机器人的行为以保证其完成预设的任务.针对该问题,目前主要借助手动推演如[9,10]与模型检测技术[11]以保证自主移动机器人算法或协议满足一定的性质[5,6].然而手动推演不仅过程冗长复杂,推演过程中也容易出现错误. 尤其对于异步调度模型,存在快照过时的情况,根本无法使用手动推演的方式进行推演验证[12].形式化方法因其自动性,严谨性与高效性逐渐被用于各种自主移动机器人协议的验证. 如 Bétrice 等人使用 DiVinE 和 ITS 工具实现了移动机器人算法的验证[12].Ha 等人使用 Maude 重写逻辑语言实现移动机器人永恒探索算法的验证[13].已知的这些方法多针对某种特定的调度模型进行建模,并通过人为设定一个具体的初始状态进行验证.然而自主移动机器人算法或协议中系统的初始状态多是未知的.尽管理论上可以通过穷举所有可能的初始状态对系统进行一一验证,然而在实际操作中不仅效率低下并容易发生遗漏而导致验证不完整.Aminof 等人提出一种参数化的模型检测方法用于聚集系统(rendezvous systems)[14].然而由于参数化模型检测的不可判定性[15],该方法只能用于某类特殊模型的验证,通常需要通过抽象(abstraction)或者归纳(induction)等手段对模型进行适当的转化使其模型检测问题变的可判定[16,17].

本文以自主机器人永恒空间探索协议为例,提出一种符号模型检测方法[18]用于自主移动机器人算法或协议的建模与验证.在三种典型的调度策略下,利用 nuXmv 工具[19]分别对机器人自主空间探索算法进行建模并利用 LTL 公式对机器人的移动行为进行定义,利用 nuXmv 提供的基于 BDD 方法[11]及 SMT 方法[11]实现了协议的符号模型检测,分别验证不同场景下(不同环节点个数与机器人个数)协议是否满足永恒探索的需求.当出现不满足情况时,nuXmv 给出不满足的状态路径作为反例,分析性质不被满足的具体原因.验证结果与 Bétrice 和 Ha 等人的验证结果相同.相比 Bétrice 和 Ha 等人的验证方法,本文提出的方法具有如下优点: 1、基于符号模型检测的方法不依赖于某个具体的初始状态,可以在不提供初始状态的前提下对协议的性质进行验证;2、符号模

.

型检测建模方法具有模块化特性,主要模块可在不同的调度模型下重复使用,大大简化了建模的复杂性;3、符号模型检测的高效性可有效避免验证过程中的状态爆炸问题.

文章结构:第 1 节介绍移动机器人探索算法的基本原理和调度策略;第 2 节着重介绍了移动机器人在环形空间上的永恒探索算法;第 3 节描述永恒探索算法在不同调度策略下在 nuXmv 中的建模;第 4 节给出相关性质的模型检测验证结果及分析;最后,第 5 节对本文工作进行总结并对未来工作做简单探讨.

1 移动机器人探索算法和调度策略的介绍

本节介绍机器人在无向连通图上的移动原理以及在不同的调度策略下探索算法的差异.

1.1 探索空间定义

探索空间一般可抽象为一个无向连通图,图的每个结点表示空间上机器人可达的位置,边表示机器人可以通过的路径,机器人沿着该条路径到达相邻的空间位置.每个结点在同一时间只能有一个机器人,黑色结点说明该空间位置在此刻有一个机器人,白色结点说明该空间位置上没有机器人.

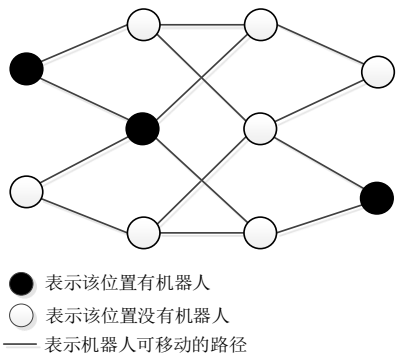

图 1 无向连通图表示探索空间

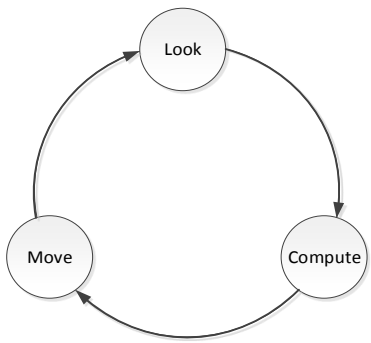

图 2 机器人移动三阶段

图 1 给出了一个简单的离散探索空间的无向连通图表示,在该探索空间上有两个机器人,它们分别位于两个黑色的空间结点上.类似于计算机网络的网络拓扑结构,探索空间结构也有总线拓扑结构、星型拓扑结构、环形拓扑结构、树形拓扑结构等.

1.2 机器人移动三个阶段

机器人移动分为三个阶段,分别是观察(look)、计算(compute)和移动(move).在观察阶段,机器人通过视觉传感器获取环境中其他机器人的位置的快照信息.计算阶段根据观察阶段获取的位置快照信息计算得出移动决策,即机器人是否移动,若移动是沿着那条路径进行.移动阶段机器人的动力装置依据移动决策作出相应的移动.如图二所示,这三个阶段是按照观察、计算、移动再到观察重复进行的.

1.3 机器人移动调度策略

假设在一个探索空间中存在$k\ (k \in \mathbb{N}^+)$个机器人,使用集合$Rob = \{r_1, r_2, \dots, r_k\}$表示.空间所有位置进行编号,使用集合$Pos = \{0,1,\dots,n-1\}$表示所有位置结点的编号.机器人与位置之间的关系使用函数$p: Rob \to Pos$表示,机器人r的位置$p(r) \in Pos$,且满足对于任意两个机器人$r_i, r_j (i \neq j)$, $p(r_1) \neq p(r_2)$.

集合 Rob 中只有被随机调度器选中的机器人,才执行移动.那么被选中的机器人非空集合$Sched \subseteq Rob$,对于$\forall r \in Shced$,执行观察、计算、移动,$\forall r \notin Shced \wedge r \in Rob$不做相关动作,这就是调度策略.目前有三种调度策略,分别是完全同步调度策略 FSYNC,、半同步调度策略 SSYNC 与完全异步调度策略 ASYNC.

在半同步调度策略中,非空集合$Sched \subseteq Rob$,任意机器人$r \in Sched$同步执行观察、计算、移动,完成这三步之后,就是完成一个移动阶段.进入下一个移动阶段之前,调度器又会随机选中一组机器人,重复执行上述的移动阶段,也就是每个移动阶段选中的集合 Sched 都是随机的.如图三,算法表示半同步调度策略具体执行过程.

其中 synchronous 块,表示同步块中所有机器人同步完成移动阶段.

FSYNC-SCHEDULE(Rob)	SSYNC-SCHEDULE(Rob)	ASYNC-SCHEDULE(Rob)

1 while
2 synchronous{
3 foreach r in Rob{
4 r.look
5 r.compute
6 r.move
7 }
8 }
9
	1 while
	2 choose Sched from Rob
	3 synchronous {
	4 foreach r in Sched{
	5 r.look
	6 r.compute
	7 r.move
	8 }
	9 }
		1 foreach r in Rob
		2 synchronous {
		3 foreach r in Rob{
		4 r.look
		5 r.compute
		6 r.move
		7 }
		8 }
		9

(a) 完全同步调度策略	(b) 半同步调度策略	(c) 完全异步调度策略

表 1 完全同步,半同步与异步调度策略

完全同步调度策略是半同步调度策略中很特殊的一种,每个移动阶段选中的机器人集合Sched = Rob,所有机器人都被调度器选中.完全同步调度策略算法如图四所示,每个移动阶段被调度器选中的是全部机器人,同步执行移动阶段.

完全异步调度策略,每个机器人异步执行移动阶段,也就是说某个机器人还在观察阶段,其他机器人或许在执行计算或者移动.在异步调度策略中,机器人会使用过时的快照信息,做出移动策略.如图五所示,每个机器人都各自执行观察、计算、移动,机器人之间是并行执行移动阶段.

2 移动机器人环形空间永恒探索算法

探索空间结构有总线拓扑结构、星型拓扑结构、环形拓扑结构、树形拓扑结构,在此以环形拓扑结构为研究对象.首先介绍环形拓扑结构环上的机器人视觉快照、匹配移动算法获取移动决策、移动决策的执行.在此基础上,将介绍永恒探索移动算法的相关概念.

2.1 机器人视觉快照

图 3 给出一个简单的环形拓扑结构探索空间,沿着环的顺时针,给每个位置结点进行递增编号,所有位置的编号组成的集合为 Pos,黑色结点表示该位置有一个机器人,白色结点表示该位置没有机器人,下面给出结点上是否有机器人的定义.

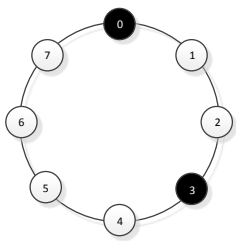

图 3 环形拓扑结构探索空间

定义 1 (结点机器人数) 结点机器人d_j,j 表示结点的编号.当结点 j 上有 k 个机器人时d_j = k,当结点上没有机器人时,$d_j = 0$.

在环形拓扑结构的空间中,机器人可以顺时针观察,也可以逆时针观察.那么每个机器人的观察位置快照就有顺时针快照和逆时针快照两种.

定义 2 (机器人快照) 结点上机器人通过视觉传感器获取位置快照信息$\delta_p^{\mathcal{F}}, \mathcal{F} \in \{+,-\}$表示快照方向,+表示

.

顺时针,-表示逆时针, $p \in$ Pos表示结点位置.在拥有 n 个结点的环形拓扑结构探索空间上,结点 j 上机器人顺时针和逆时针的快照如下:

$$\text{顺时针序列定义: } \delta_j^+ = \langle \mathrm{d}_j, \mathrm{d}_{j+1,\ldots,} \mathrm{d}_{\mathrm{j}+n-1} \rangle$$
$$\text{逆时针序列定义: } \delta_j^- = \langle \mathrm{d}_j, \mathrm{d}_{j-1,\ldots,} \mathrm{d}_{j-\mathrm{n}+1} \rangle$$

如图 3 中结点 0 为例,其顺时针和逆时针快照如下:

$$\text{顺时针序列 1: } \delta_0^+ = \langle 1,0,0,1,0,0,0,0 \rangle$$
$$\text{逆时针序列 1: } \delta_0^- = \langle 1,0,0,0,01,0,0 \rangle$$

这种机器人快照表示方法有一定弊端,当环形拓扑结构探索空间结点数 n 很大时,这种表示方式很不方便.后续提出了一种 F-R 的定义方式,这里是可以表示连续无机器人结点个数和连续有机器人结点个数.F_m表示连续 m 个结点无机器人,R_n表示连续 n 个结点有机器人.那么顺时针序列 1 和逆时针序列 1 转化成 F-R 的定义方式为:

$$\text{顺时针 F}-\text{R 序列 1: } \delta_0^+ = \langle R_1, F_2, R_1, F_4 \rangle$$
$$\text{逆时针 F}-\text{R 序列 2: } \delta_0^- = \langle \mathrm{R}_1, \mathrm{F}_4, \mathrm{R}_1, \mathrm{F}_2 \rangle$$

2.2 环形拓扑结构空间移动机器人探索算法

环形拓扑结构空间的探索模式主要有两种,即环形空间探索停止(Ring exploration with stop)和环形空间永恒探索(Perpetual ring exploration).本文重点研究环形空间永恒探索算法.

2.2.1 最小移动算法

最小移动算法是指环形拓扑结构空间结点数为 n ≥ 10,环形拓扑结构空间上机器人数 r = 3,并且 n 和 r 数量关系互质的情况下,确保机器人的移动满足环形空间永恒探索.移动算法可分为两个阶段,分别是稳定阶段(Legitimate phase)和收敛阶段(Convergence phase).

最小移动算法中的稳定阶段的移动规则定义如下:

Legitimate phase			
RL1::	$\delta_{c(r)}^{\mathcal{F}} = \langle R_2, F_2, R_1, F_{n-5} \rangle$	→	r.Back
RL2::	$\delta_{c(r)}^{\mathcal{F}} = \langle R_1, F_1, R_1, F_{n-6}, R_1, F_2 \rangle$	→	r.Front
RL3::	$\delta_{c(r)}^{\mathcal{F}} = \langle R_1, F_3, R_2, F_{n-6} \rangle$	→	r.Front

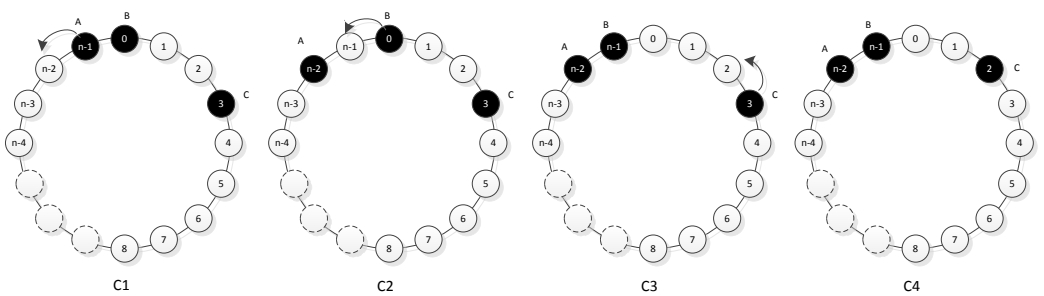

图 4 最小移动算法的稳定阶段

如图 4 所示,C1 中机器人 A 快照顺时针快照$\delta_{\mathrm{c(A)}}^{+} = \langle \mathrm{R}_2, \mathrm{F}_2, \mathrm{R}_1, \mathrm{F}_{n-5} \rangle$,匹配移动算法 RL1 做出后退的决策,其他机器人 B、C 此时无匹配移动算法,保持静止,只有机器人 A 从结点 n-1 移动到结点 n-2.相同,在 C2 中机器人 B 从结点 0 移动到结点 n-1,C3 中机器人 C 从结点 3 移动到结点 2.最后如 C4 所示,对于机器人 A、B、C 而言,在 C1 和 C4 状态下所获取的快照是一样的,也就是说,整个系统会重复上述 C1 到 C4 的过程.此时,系统被称

为进入了稳定阶段.稳定阶段的系统状态称为稳定状态.系统进入稳定状态,就会永远停留在稳定阶段.

系统除了稳定状态之外,还有其他的状态,在这些状态下,系统通过一定的规则逐渐到达某个稳定状态,进而系统进入稳定状态.系统从非稳定状态到达稳定状态的过程,被称为收敛阶段.以最小移动算法为例,其收敛阶段的移动规则定义如下:

Convergence phase					
RC1::	$4 \le x \le z$	$\wedge$	$\delta_{c(r)}^{\mathcal{F}} = \langle R_1, F_x, R_2, F_z \rangle$	$\rightarrow$	r.Front
RC2::	$x \ne y$, $x > 0$	$\wedge$	$\delta_{c(r)}^{\mathcal{F}} = \langle R_1, F_x, R_1, F_y, R_1, F_x \rangle$	$\rightarrow$	r.Doubt
RC3::	$0 < x < z < y \wedge (x,y) \ne (1,2)$	$\wedge$	$\delta_{c(r)}^{\mathcal{F}} = \langle R_1, F_x, R_1, F_y, R_1, F_z \rangle$	$\rightarrow$	r.Front
RC4::			$\delta_{c(r)}^{\mathcal{F}} = \langle R_3, F_{n-3} \rangle$	$\rightarrow$	r.Back
RC5::			$\delta_{c(r)}^{\mathcal{F}} = \langle R_1, F_1, R_2, F_{n-4} \rangle$	$\rightarrow$	r.Back

2.2.2 环形空间永恒探索算法的性质

在环形拓扑结构空间中,要求在任意时刻每个结点至多只有一个机器人,并且对于任何一个机器人无论其在初始状态位于环上哪个节点,都可以无限次的访问环上的任意一个节点.这两个性质分别被称为非冲撞性与不终止性.另外,对任意两个物理位置上相邻的机器人,不会发生两个机器人在移动过程中相互碰撞的情况,即两个机器人不会同时互相交换位置.

- **非冲撞性**:同一时刻每个结点上,至多有一个机器人,两个机器人不能同时经过同一条边.
- **非互换性**:.物理上相邻的两个机器人不会同时互相交换位置.
- **非终止性**:每个机器人对都可以对环上任意一个结点无限次地的访问.

3 基于 nuXmv 的空间永恒探索算法的建模

使用 nuXmv 对移动机器人空间永恒探索算法分别根据完全同步调度策略 FSYNC、半同步调度策略 SSYNC、完全异步调度策略 ASYNC 三种调度策进行建模,根据建立的模型验证机器人移动算法是否满足永恒探索的性质.本章节主要以最小移动算法是否是永恒探索算法为例,详述建模与验证过程.

3.1 nuXmv的简单介绍

nuXmv 是经典符号模型检测工具 nuSmv 的后续版本,用于有限状态和无限状态系统(尤其是同步系统)的验证分析,支持 CTL 及 LTL 等时序逻辑公式的符号模型检测.对于有限状态系统,采用当前最先进的 SAT 求解算法,而对于无限状态系统,通过基于 SMT 可满足性求解利用 MathSAT5[20]求解工具实现性质的验证.

nuXmv 继承了 nuXmv 模块化建模特性.每个模块(Module)可描述一个状态迁移系统.模块主要包含三个部分:变量(variables),约束(constraints)和规范(specification).变量用于定义系统状态,约束用于定义状态转移关系及模型的一些约束条件,而规范表示利用 CTL 或者 LTL 逻辑公式定义的将要验证的系统性质.对于包含多个具有相同行为实体的系统,可以定义一个参数化的模块描述实体的行为,然后通过模块实例化描述系统的整体行为.因此,nuXmv 模块化特性使其非常适用于自主移动机器人系统的建模.此外,nuXmv 符号模型检测方式可以在不提供系统初始状态的条件下对系统进行验证.而自主移动机器人系统大多不对机器人的具体的初始位置进行设定,而只定义一些必要的约束,如每个节点上至多只有一个机器人等.同时该特性还可以有效避免系统的状态爆炸问题.因此,从建模与验证的角度分析,nuXmv 都适用于自主移动机器人系统的形式化验证与分析.

3.2 移动机器人空间永恒探索算法的建模

3.2.1 移动机器人的建模

利用 nuXmv 提供的参数化模块(parameterized module)对算法中的移动机器人建模.模块的主要部分如图 5 所示.模块带有三个参数 p1,p2 和 p3,分别表示机器人自身机器左右两个相邻机器人的位置.需要相邻机器人作为参数的原因是机器人的移动需要考虑其相邻机器人的位置.机器人的状态用两个变量 `phase` 和 `move` 表示.

.

```
1  MODULE robot(p1,p2,p3)     -- p1,p2,p3 表示机器人及其两个邻居的位置
2    VAR
3      phase : {lc,m};         -- lc 表示 look&computing, m 表示 move
4      move  : -1..1;          -- 变量 move 的值可为-1, 0 和 1
5    ASSIGN
6      init(phase) := lc;      -- 变量 phase 的初始值为 lc, move 为 0
7      init(move)  := 0;       -- 注意机器人的初始位置并未确定
8      next(phase) :=          -- 以下 6 行表示变量 phase 在后续状态中的值
9        case
10         phase = lc    :  m;      -- 若当前值为 lc, 则在下一个状态为 m
11         phase = m     :  lc;     -- 若当前值为 m, 则在下一个状态为 lc
12         TRUE          :  phase;  -- 否则,phase 的值不变
13       esac;
14     next(move) :=     -- 变量 move 在下一步的值.仅以规则 RL1 情况为例
15       case
16         phase = lc & (p2-p1+10) mod 10 = 1 & (p3-p2+10) mod 10 = 3 : -1;
17         phase = lc & (10-(p3-p1+10) mod 10) mod 10 = 1 & (10-(p2-p3+10) mod
18         10) mod 10 = 3 : 1;
19         …
20       esac;
21     next(p1) :=      -- 机器人在下一个状态是的位置
22       case
23         phase = m & (move + p1) <= 0                      : 10 + (move + p1);
24         phase = m & (move + p1) > 0 & (move + p1) <= 10: (move + p1);
25         phase = m & (move + p1) > 10                       : (move + p1) - 10;
26         TRUE                                              : p1;
27       esac;
28   FAIRNESS       -- 公平性
29     Running
```

图 5 利用 NuXmv 对移动机器人的定义模块

前者表示机器人的移动状态,即查看-计算阶段或移动阶段.后者表示机器人的移动距离,取值可为-1,1 和 0,分表表示向后,向前移动一个节点和不发生移动.

以下以变量 move 的值得计算和机器人位置 p1 的变化为例, 重点介绍机器人移动过程的建模. 如图 5 中第 16 至 18 行代码所示,使用 NuXmv 描述移动算法 RL1 顺时针方向的匹配,在 p1 位置上的机器人 r 从自身开始,按照顺时针方向,依次计算 p1 到 p2、p2 到 p3 之间的间隔个数.对应匹配 RL1 中机器人的之间的间隔个数,以此作为机器人 r 顺时针匹配移动算法 RL1 的依据.逆时针方向的匹配则是计算 p1 到 p3、p3 到 p2 之间的间隔个数作为匹配依据.相邻机器人 A 到机器人 B 间隔个数使用$gap_{A\to B}^{\mathcal{F}}$表示,其中$\mathcal{F}\in\{+,-\}$表示方向,+表示顺时针,−表示逆时针.机器人 r 的移动量 move 使用M_r表示.

下面介绍$gap_{A\to B}^{\mathcal{F}}$和$M_r$的计算方法.$gap_{A\to B}^{\mathcal{F}}$的定义如下:

$$gap_{A\to B}^{+}=(c(B)-c(A)+n)\ mod\ n \qquad \text{(顺时针)}$$

$$gap_{A\to B}^{-}=\Big(n-\big((c(B)-c(A)+n)\ mod\ n\big)\Big)\ mod\ n \qquad \text{(逆时针)}$$

其中,n表示环上节点的个数.在环形拓扑结构空间中,每个机器人在空间每个时刻都有一个位置,可以根据机器人位置计算机器人之间的间隔个数,这样可以知道每个机器人顺时针和逆时针的快照 F-R 序列. 如图 6 中所示,环形拓扑结构空间都按照顺时针方向进行位置编号,位置 a 处是机器人 A,位置 b 处是机器人 B,当结点编号到 n-1 时,下一个位置是 0,那么计算机器人 A 到机器人 B 之间连续没有机器人的结点数量时,就要考虑机器人 A 和机器人 B 之间是否包含位置为 0 的情况.B 与 A 之间的顺时针间隔为两者节点位置的差加n再以n为基数取模.逆时针的间隔则为n减去顺时针的间隔得到的结果再以n为基数取模.之所以继续取模是考虑 A 与 B 位置重

叠的情况,此时,顺时针与逆时针方向的间隔均为 0.

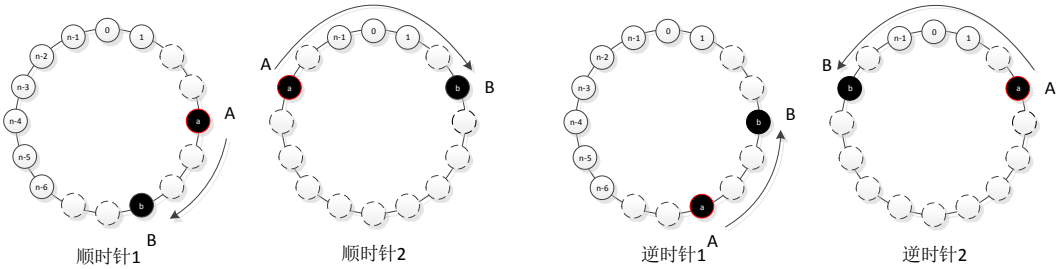

图 6 顺时针和逆时针快照获取

假设机器人 r 的 F-R 快照$\delta^{\mathcal{F}}_{c(r)}$满足移动算法中的某条规则 L,记为$match(\delta^{\mathcal{F}}_{c(r)}, L)$,则根据规则 L 可确定 r 的移动方向,用变量$\beta_L$表示,从而计算出 r 的移动量$M_r$.计算公式如下:

$$M_r = \begin{cases} 1 \;\; if \;\; \left(match\left(\delta^{\mathcal{F}}_{c(r)}, L\right) \wedge \mathcal{F} = + \;\wedge\; \beta_L = Front\right) \vee \left(match\left(\delta^{\mathcal{F}}_{c(r)}, L\right) \wedge \mathcal{F} = - \;\wedge\; \beta_L = Back\right) \\ -1 \;\; if \;\; \left(match\left(\delta^{\mathcal{F}}_{c(r)}, L\right) \wedge \mathcal{F} = + \;\wedge\; \beta_L = Back\right) \vee \left(match\left(\delta^{\mathcal{F}}_{c(r)}, L\right) \wedge \mathcal{F} = - \;\wedge\; \beta_L = Front\right) \\ 0 \;\; otherwise \end{cases}$$

对于环形拓扑结构空间,如果同一机器人 r 在位置c(r)的顺时针和逆时针快照 F-R 序列相同$\delta^{+}_{c(r)} = \delta^{-}_{c(r)}$,即机器人 r 的快照 F-R 序列是对称的.若 r 的快照 F-R 匹配算法中某条规则 L,则根据规则 L 得出的移动方向β_L无论是前进或者后退,上述公式中的两个条件均成立,即机器人的移动量可为 1 或者 -1,此时则随机选择一个值.若 r 的快照 F-R 不能匹配算法中任何一个移动规则时,$M_r = 0$.

机器人进入移动阶段,其下一个位置结点编号为$next(\mathrm{c(r)}) = \mathrm{c(r)} + M_r$.机器人在新的位置,改变了整个系统中机器人的快照 F-R 序列,机器人根据新的快照 F-R 序列,做出对应的移动决策.

3.2.2 三种调度策略的建模

对于完全同步调度策略和异步调度策略,利用上节对移动机器人定义的模块进行实例化.假设只考虑三个机器人的情况.用 r1,r2 和 r3 分别表示三个机器人,变量 p1,p2 和 p3 表示三个机器人在环上的位置.三个机器人的位置关系为从顺时针的方向 r1 在 r2 之后,r2 在 r3 之后.则 r1 可被定义为机器人模块的一个实例,如表 2(a)所示.机器人 r2 与 r3 的实例化与 r1 类似.而异步调度策略的建模只需要在机器人模块实例化时加上关键字 **process**,如表 2(c)所示.该关键字的作用是声明实例之间的执行以异步方式进行[1].

对于半同步调度策略,每个机器人在调度过程中只有选中和非选中两种情况.因此在移动机器人定义模块给机器人设置一个调度变量 *dispatcher*,可以有 *chosen*(选中)与 *steady*(非选中)两种取值情况.在 nuXmv 中,在声明变量时可指定变量的取值范围.通过关键字 VAR 声明*dispatcher*变量即其取值范围,代码如下:

$$\text{VAR } dispatcher : \{chosen, steady\}$$

声明中指出*dispatcher*的取值为choose 或 steady.在使用*dispatcher*时,不给*dispatcher*赋初始值,系统初始状态值也是随机的,即 *dispatcher*的初始值可以是*chosen*也可以是*steady*.下一个移动过程开始前,都会执行一次调度,每次调度是随机的,即对于每个机器人来说,下一个移动过程中*dispatcher*的取值,在取值范围之内都是随机的.使用 nuXmv 的 *next* 函数,对机器人下一个移动过程中,*dispatcher*的取值进行定义:

$$next(dispatcher) := \{chosen, steady\}$$

1 注：nuXmv 主要针对同步系统的验证,不再支持关键字 process.使用关键字 process 将调用 nuXmv 早期版本 nuSmv 的模型检测功能.

r1 : robot(p1,p2,p3);
r2 : robot(p2,p3,p1);
r3 : robot(p3,p1,p2);
	r1 : robot′(p1,p2,p3);
	r2 : robot′(p2,p3,p1);
	r3 : robot′(p3,p1,p2);
		r1 : **process** robot(p1,p2,p3);
		r2 : **process** robot(p2,p3,p1);
		r3 : **process** robot(p3,p1,p2);

(a) 完全同步调度模型	(b) 半同步调度模型	(c) 异步调度模型

表 2 三种调度策略的建模

next 函数中,指明*dispatcher*的可取值范围,也就是说,下一个移动状态时,*dispatcher*随机取*choose, steady*其中之一.机器人在匹配移动算法时,若*dispatcher*变量取值为*choose*,才做移动决策的计算.

新的移动机器人定义的模块被命名为 `robot′` ,与 `robot` 具有相同的三个参数.为描述半同步调度策略,只需要利用模块 `robot′`初始化机器人实例即可,如表 2 (b)所示.

3.3 永恒探索性质LTL公式定义

根据上节对移动机器人空间永恒探索算法的建模,算法的无冲撞性,非互换性和非终止性可分别用如下如下 LTL 公式定义:

$$\bigwedge_{i=1}^{k}\bigwedge_{j=1}^{k}\square\left(i\neq j\Rightarrow c(r_i)\neq c(r_j)\right) \quad \text{(非冲撞性)}$$

$$\bigwedge_{h=0}^{n-1}\bigwedge_{i=1}^{k}\bigwedge_{j=1}^{k}\square\left(c(r_i)=h\wedge c\left(r_j\right)=(h+1)\ mod\ n\Rightarrow X\left(\neg(c(r_i)=(h+1)\ mod\ n)\wedge c\left(r_j\right)=h\right)\right) \quad \text{(非互换性)}$$

$$\bigwedge_{h=0}^{n-1}\bigwedge_{i=1}^{k}\square\Diamond\left(c(r_i)=h\right) \quad \text{(非终止性)}$$

非冲撞性对应的 LTL 表示在任何时刻任意两个不同的机器人r_i与r_j其在环上的位置一定不同,即等同于在任何节点上至多不会超过一个机器人.非互换性对应的 LTL 公式表示如果两个机器人r_i与r_j物理上相邻,即其所在节点的编号相差 1,则一定不会出现r_i与r_j在下一个状态位置互换的情况.非终止性对应的公式表示任意一个机器人r_i均可无限次的访问环上任何一个节点.

上述三个性质都以移动机器人空间永恒探索算法满足公平性为前提,即任何一个机器人都可以被无限次的调度,公平性可用如下 LTL 公式描述:

$$\bigwedge_{i=1}^{k}\square\Diamond\left(r_i.running=true\right) \quad \text{(公平性)}$$

4 验证结果与分析

根据上节定义的移动机器人空间永恒探索算法模型,利用 nuXmv 工具对算法的非冲撞性、非互换性和非终止性在不同的调度策略下进行模型检测.实验运行环境为 Windows10 专业版,硬件环境为 CPU Intel Xeon(R) 3.40GHz,16G 内存.表3给出了三个性质在不同场景下的验证结果以及所需时间.实验中考虑了节点个数分别为 10 到 17 的情况，其中由于算法要求机器人个数与节点个数互质,节点为 12 与 15 的情况在实验中略去.验证结果表明移动机器人空间永恒探索算法在完全同步模型和半同步模型下满足对环形空间永恒探索性质(非终止性),此结果与 Béatrice 等人的结论是一致的[12].此外,也验证了算法同样满足非碰撞性及非互换性.

而在异步调度模型下,验证的结果为算法不满足非碰撞性与非终止性,nuXmv 返回相应的反例.根据反例分析,永恒探索性质不能被满足的原因是异步过程中机器人使用过时的快照信息做出的移动决策会导致相邻的机器人发生碰撞,此时同一个位置结点上有两个机器人,后续所有机器人都没有与移动算法相匹配,导致所有的机器人都不能移动.由此可见,非终止性不被满足的原因是非碰撞性没有被满足.这一结果与 Ha 等人利用 Maude 模型检测得出的反例相同.在其反例中,用于出现了两个机器人碰撞的情况而导致所有机器人无法移动,Ha 等人将此情况称为死锁状态.与 Ha 等人的验证不同的是其方法需要给出具体的初始状态才发现了反例,而如何发现导致反例的初始状态文章并没有交待.本文提出的方法可以在不给出初始状态的前提下依然找到对应的反例.

节点数	完全同步调度模型						半同步调度模型						异步调度模型					
	非碰撞性		非终止性		非互换性		非碰撞性		非终止性		非互换性		非碰撞性		非终止性		非互换性	
	结果	耗时	结果	耗时	结果	耗时	结果	耗时	结果	耗时	结果	耗时	结果	耗时	结果	耗时	结果	耗时
10	✓	1.4s	✓	484.0s	✓	1.9s	✓	0.7s	✓	83.1s	✓	1.7s	×	1.2s※	×	14.7s※	✓	0.9s
11	✓	5.4s	✓	3181.1s	✓	4.3s	✓	1.1s	✓	723.1s	✓	4.3s	×	9.3s※	×	18.1s※	✓	0.7s
13	✓	8.3s	✓	42.4s*	✓	6.8s	✓	1.1s	✓	63.9s*	✓	6.7s	×	10.7s※	×	2.1s※	✓	0.6s
14	✓	8.9s	✓	90.4s*	✓	8.4s	✓	1.1s	✓	129.2s*	✓	8.8s	×	9.7s※	×	48.8s※	✓	0.6s
16	✓	1.2s	✓	168.4s*	✓	2.3s	✓	0.6s	✓	389.2s*	✓	2.4s	×	4.9 s※	×	19.6s※	✓	0.7s
17	✓	13.0s	✓	315.7s*	✓	11.4s	✓	2.0s	✓	562.3s*	✓	12.2s	×	27.5s※	×	102.2s※	✓	0.6s

注: ✓表示验证结果为真;×表示验证结果为假;*表示在设置初始状态的情况下验证所需时间;-表示超时;
※表示采用 SMT 方法验证所需时间;未标注的时间均表示采用 BDD 方法所需的验证时间.

表 3 三种调度策略下的模型检测结果

表 3 也反映出符号模型对移动机器人空间永恒探索算法验证的高效性.nuXmv 支持基于 BDD 和 SMT 方法的验证.对于验证性质的正确性,BDD 方法的效率相对较高,而对于不被满足的性质,SMT 方法可以更快的找到反例.在实验中采用两种方法对三个性质进行验证.结果表明大部分验证都可以在相对较短的时间内完成.虽然随着空间节点数的增长验证所需的时间也会有所增加,但依然可以在较合理的时间如一小时内完成.同 Béatrice 与 Ha 等人的工作相比,nuXmv 找到反例的时间更短.然而在验证被算法满足的性质时,nuXmv 所需的时间相对较长,这是因为 nuXmv 不需要设定具体的初始状态,因此其搜索的状态空间比固定初始状态时更大,所需的时间则较长.表 3 同样给出在固定初始状态的情况下验证所需的时间,数据表明时间明显缩短..

5 总结以及未来工作

本文提出了自主移动机器人系统的符号化模型检测方法,以移动机器人空间永恒探索算法为例,介绍了算法在完全同步,半同步及异步调度模式下的形式化建模与验证过程,分别验证了算法的不冲突性,不终止性及不交换性等主要性质.验证结果表明在不提供初始状态的条件下通过符号化模型检测的方法同样可以找到算法在异步调度模式下不满足不终止性的反例.实验结果表明符号模型检测方法从建模和验证方面都适用于自主移动机器人系统的验证.

尽管符号模型检测方法可以有效避免状态爆炸问题,然而实验结果发现对异步调度模型随着节点个数的增加模型检测的时间也在快速增长.因此,本文提出的方法无法直接用于实际系统中节点个数较多的情景.在本方法的基础上,通过抽象或者归纳技术,如 nuXmv 内置的 k-induction 方法[19],以解决验证的效率问题,是将来需要进一步深入研究的工作.

References:

[1] Bonnet F, Défago X, Petit F, et al. Brief Announcement, Discovering and Assessing Fine-Grained Metrics in Robot Networks Protocols. Stabilization, Safety, and Security of Distributed Systems. Springer Berlin Heidelberg, 2012:282-284.

[2] Suzuki I, Yamashita M. Distributed Anonymous Mobile Robots: Formation of Geometric Patterns. Sirocco'96, the, International Colloquium on Structural Information & Communication Complexity, 1999:313-330.

[3] Suzuki I, Yamashita M. Erratum: Distributed anonymous mobile robots: formation of geometric patterns. Siam Journal on Computing, 1999, 28(4):1347-1363.

[4] Flocchini P, Prencipe G, Santoro N. Distributed Computing by Oblivious Mobile Robots. Morgan & Claypool Publishers, 2012

[5] Blin L, Milani A, Potop-Butucaru M, et al. Exclusive Perpetual Ring Exploration without Chirality. Lecture Notes in Computer Science, 2010, 6343:312--327.

.

[6] Flocchini, Paola, Ilcinkas, et al. Computing Without Communicating: Ring Exploration by Asynchronous; Oblivious Robots. Algorithmica, 2013, 65(3):562-583.

[7] Kamei S, Lamani A, Ooshita F, et al. Asynchronous Mobile Robot Gathering from Symmetric Configurations without Global Multiplicity Detection. Lecture Notes in Computer Science, 2011, 6796:150-161.

[8] Flocchini P, Prencipe G, Santoro N, et al. Gathering of asynchronous robots with limited visibility. Theoretical Computer Science, 2005, 337(1):147-168.

[9] Baldoni R, Bonnet F, Milani A, et al. On the Solvability of Anonymous Partial Grids Exploration by Mobile Robots. Principles of Distributed Systems. Springer Berlin Heidelberg, 2008:428-445.

[10] Devismes S, Lamani A, Petit F, et al. Optimal Grid Exploration by Asynchronous Oblivious Robots. Symposium on Self-Stabilizing Systems. Springer Berlin Heidelberg, 2012:64-76.

[11] Clarke, E., Grumberg, O., Peled, D.: Model Checking. MIT Press, Cambridge, 2001.

[12] Bérard B, Lafourcade P, Millet L, et al. Formal verification of mobile robot protocols. Distributed Computing, 2013:1-29.

[13] Doan H T T, Bonnet F, Ogata K. Model Checking of a Mobile Robots Perpetual Exploration Algorithm. International Workshop on Structured Object-Oriented Formal Language and Method. Springer, Cham, 2016:201-219.

[14] Aminof, B., Kotek, T., Rubin, S., Spegni, F., Veith, H.: Parameterized model checking of rendezvous systems. In: Paolo, B., Daniele, G. (eds.) CONCUR 2014 Concurrency Theory, vol. 8704 of Lecture Notes in Computer Science, Springer, Berlin, 2014: 109–124

[15] Apt, K.R., Kozen, D. Limits for automatic verification of finitestate concurrent systems. Inf. Process. Lett. 1986,22(6):307–309.

[16] Clarke, E.M., Grumberg, O., Jha, S. Veryfying parameterized networks using abstraction and regular languages. In: Proceedings of 6th International Conference on Concurrency Theory (CONCUR'95), 1995, 962:395–407.

[17] Manna, Z., Pnueli, A. Temporal verification diagrams. In: Proceedings of International Conference on Theoretical Aspects of Computer Software (TACS'94), 1994, 789:726–765

[18] Clarke E M, Mcmillan K L, Hartonas-Garmhausen V. Symbolic Model Checking. International Conference on Computer Aided Verification. Springer-Verlag, 1996:419-427.

[19] Cavada R, Cimatti A, Dorigatti M, et al. The nuXmv Symbolic Model Checker. Computer Aided Verification. Springer International Publishing, 2014:334-342.

[20] Cimatti A, Griggio A, Schaafsma B J, et al. The MathSAT5 SMT Solver. International Conference on TOOLS and Algorithms for the Construction and Analysis of Systems. 2013:93-107.