软件学报ISSN 1000-9825, CODEN RUXUEW E-mail: jos@iscas.ac.cn

*Journal of Software*, [doi: 10.13328/j.cnki.jos.000000] http://www.jos.org.cn

©中国科学院软件研究所版权所有. Tel: +86-10-62562563

自治移动机器人永恒探索算法的符号化模型检测方法[[1]](#footnote-1)\*

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 计算机科学与软件工程学院,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,自治移动体(autonomous mobile agents)在网络中的作用也日益重要,利用形式化的方法验证自治移动体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.本文以自治移动机器人探索算法为例,提出自治移动体符号化模型检测方法,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,借助符号化模型检测的高效性,有效避免了状态爆炸问题.利用NuXmv符号化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（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化模型检测对自治移动机器人算法形式化验证的高效性。

关键词: nuXmv; LTL;移动机器人算法;模型验证;

中图法分类号: TP311

中文引用格式: 蔡晓伟，张民. 基于nuXmv的移动机器人永恒探索算法的形式化验证与分析. 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. A nuXmv-Based Approach to Formal Verification and Analysis of Mobile Robots Perpetual Exploration Algorithm. Symposium on Dependable Software Engineering, SETTA 2017 (in Chinese).http://www.jos.org.cn/1000-9825/0000.htm

A nuXmv-Based Approach to Formal Verification and Analysis of Mobile Robots Perpetual Exploration Algorithm

CAI Xiaowei1,2, ZHANG Min1,2

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

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

**Abstract**: In the field of mobile robot exploration of unknown environment space, the study found some robot exploration algorithm. However, in the course of the study, most scholars use the method of manual deduction to verify the correctness and rigor of the algorithm. This method of verification is not efficient, the deduction process is cumbersome, and it is prone to error. In order to verify the robot algorithm more quickly and accurately, this paper proposes the verification and analysis of the robot exploration algorithm using the nuXmv formal model validation tool. The nuXmv tool supports verifying the validation of the LTL formula, using the LTL formula to describe any initial state of the robot, and using the LTL to describe the suitability of the robot to explore the algorithm. NuXmv proposed three models of full synchronization model (FSYNC), semi-synchronous model (SSYNC), and asynchronous model (ASYNC). The model can be verified by the algorithm model. The search algorithm does not satisfy the eternal exploration of the nature, nuXmv can give the state of the state is not satisfied, the analysis is very convenient.

**Key words**: nuXmv; LTL; Mobile robots; Verification; Model checking

移动机器人所能完成任务越来越多，任务复杂程度不断增加。许多应用场景设想移动机器人可以在没有任何中央调度的情况下，机器人可以互相协调合作完成任务，这种多机器人系统可以应用于一些特殊环境，比如未知区域的地图构造、环境监测、城市搜索、在危险，人员无法进入的区域进行抢救，监视、未知空间的探索等。在这些应用场景中，自主移动机器人系统的可靠性至关重要，意味着在使用自主移动机器人完成这类任务之前，需要正确和严密验证其可靠性。

本文的研究问题是机器人自主空间探索算法验证方法的改进，机器人自主空间探索算法也可以称为探索协议(exploration protocols)，其中有两种变体：探索终止(exploration with stop)和永恒探索(perpetual exclusive exploration)。在详述探索算法之前，介绍一下机器人的属性。

所有机器人是完全相同，机器人之间无法通过外表进行识别，并且都执行相同的探索算法。并且机器人自身具备以下几个特征：1、无记忆存储：不能存储过去完成的动作。2、无方向传感器：本身无法辨别方向，并且没有偏好移动方向。3、机器人之间无通讯功能：机器人之间不能发送和接收消息。4、但是，机器人有视觉传感器：可以通过视觉传感器获取空间其他机器人的位置信息。

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，图的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，更加关注的是机器人的数量和空间位置之间的关系。但是，在验证某些离散空间(例如：环)的探索算法时，也需要考虑对称性的验证问题。

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，for the fully-synchronous model)和半同步调度模型(SSYNC,semi-synchronous model)。随后，由Flocchini et al提出了异步调度模型(ASYNC，asynchronous, in the sequel)，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，每个机器人异步执行移动阶段，也就是说，此时机器人移动可能会使用过时的快照信息，做出移动策略。前面的两种调度策略下，手动验证机器人探索算法，过程十分冗长、复杂，容易出错、遗漏。异步调度模型，由于是异步，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。

本文的贡献在使用形式化模型验证方法验证永恒探索算法在完全同步调度模型、半同步调度模型、异步调度模型三种模型下的满足性。使用nuXmv描述调度策略模型和探索算法，LTL公式描述机器人未来移动位置的可满足性，验证探索算法是否满足永恒探索要求。当出现不满足情况时，nuXmv可以给出不满足的状态路径，可以清晰分析不满足时的具体细节。近期，Béatrice Bérard的一篇文章中使用DiVinE和ITS工具实现了移动机器人算法的验证，同时Ha Thi Thu Doan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。本文提出使用nuXmv的验证方法，验证相同的探索算法时，结果与他们是一致的，nuXmv在时间效率上和Béatrice Bérard所使用的方法相差无几。与前面工作不同的有以下几点：1、nuXmv可以使用LTL公式描述机器人在图上任意初始状态，验证在任意初始状态下探索算法的可满足性。2、使用nuXmv描述探索算法相对比较直观。

文章结构：第一章移动机器人永恒探索算法和调度策略介绍；第二章nuXmv探索算法在不同调度策略下的建模；第三章探索算法的模型检测与结果分析；第四章本文工作与相关工作的对比分析；第五章本文工作总结与未来工作的展望。

# 移动机器人探索算法和调度策略的介绍

本章介绍机器人在图上移动过程，机器人在不同的调度策略下探索算法的差异。为后续的章节nuXmv机器人探索算法建模和模型验证提供预备知识。

## 探索空间定义

使用图抽象探索空间，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。



图一，离散图表示探索空间

如图一所示，给出了一个简单的离散探索空间，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、树形拓扑结构等等。

## 机器人移动三个阶段

了解了探索空间，下面介绍一下空间结点上机器人移动执行过程。机器人移动分为三个阶段，分别是观察(look)、计算(compute)、移动(move)。



图二，机器人移动的三个阶段

如图二所示，描述机器人三个阶段，对于机器人来说，这三个阶段是按照观察、计算、移动再到观察重复进行的。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。

## 机器人移动调度策略

在一个探索空间中存在个机器人，使用集合表示。空间所有位置进行编号，使用集合表示所有位置结点的编号。机器人与位置之间的关系使用表示，机器人的位置。

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Rob，对于 ，执行观察、计算、移动，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,Fully Synchronous)、半同步调度策略(SSYNC,Semi Synchronous)、完全异步调度策略(ASYNC,Asynchronous)。

在半同步调度策略中，非空集合，任意机器人同步执行观察、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 | | | | 1 | while | | | | 1 | foreach r in Rob | | | |
| 2 |  | synchronous{ | | | 2 |  | choose Sched from Rob | | | 2 |  | synchronous { | | |
| 3 |  |  | foreach r in Rob{ | | 3 |  | synchronous { | | | 3 |  |  | foreach r in Rob{ | |
| 4 |  |  |  | r.look | 4 |  |  | foreach r in Sched{ | | 4 |  |  |  | r.look |
| 5 |  |  |  | r.compute | 5 |  |  |  | r.look | 5 |  |  |  | r.compute |
| 6 |  |  |  | r.move | 6 |  |  |  | r.compute | 6 |  |  |  | r.move |
| 7 |  |  | } | | 7 |  |  |  | r.move | 7 |  |  | } | |
| 8 |  | } | | | 8 |  |  | } | | 8 |  | } | | |
| 9 |  |  | | | 9 |  | } | | | 9 |  | | | |
| 完全同步调度策略 | | | | | 半同步调度策略 | | | | | 完全异步调度策略 | | | | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| SSYNC-SCHEDULE(Rob) | | | | | |
| 1 | while | | | |  |
| 2 |  | choose Sched from Rob | | |  |
| 3 |  | synchronous { | | |  |
| 4 |  |  | foreach r in Sched | |  |
| 5 |  |  |  | r.look |  |
| 6 |  |  |  | r.compute |  |
| 7 |  |  |  | r.move |  |
| 8 |  | } | | |  |

图三，半同步调度策略算法表示

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，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。

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| SSYNC-SCHEDULE(Rob) | | | | | |
| 1 | while | | | |  |
| 2 |  | synchronous { | | |  |
| 3 |  |  | foreach r in Rob | |  |
| 4 |  |  |  | r.look |  |
| 5 |  |  |  | r.compute |  |
| 6 |  |  |  | r.move |  |
| 7 |  | } | | |  |

图四，完全同步调度策略算法表示

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。

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| SSYNC-SCHEDULE(Rob) | | | | | |
| 1 | foreach r in Rob | | | |  |
| 2 |  | while | | |  |
| 3 |  |  | synchronous { | |  |
| 4 |  |  |  | r.look |  |
| 5 |  |  |  | r.compute |  |
| 6 |  |  |  | r.move |  |
| 7 |  |  | } | |  |

图五，完全异步调度策略

# 环上机器人移动算法介绍

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。为后续使用NuXmv工具建立在环形拓扑结构探索空间上的移动算法建模，提供预备知识。

## 机器人视觉快照

如图六，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。



图六，环形拓扑结构探索空间

**定义1（结点机器人数）**结点机器人，j表示结点的编号。当结点j上有k个机器人时，当结点上没有机器人时，。

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，ℱ表示快照方向，+表示顺时针，-表示逆时针，𝑝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：

如图六中结点0为例，其顺时针和逆时针快照如下：

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。表示连续m个结点无机器人，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：

## 机器人移动算法

环形拓扑结构空间的探索，提出了两种探索模式，环形空间探索停止(Ring exploration with stop)和环形空间永恒探索(Perpetual ring exploration)。在本文中主要以环形空间永恒探索为研究主体。

### 环形空间永恒探索

在环形拓扑结构空间中，每个结点至多有一个机器人，并且对于任意机器人初始位置，在机器人移动算法下，机器人探索满足以下三个性质，称该探索为环形空间永恒探索，移动算法称为永恒探索算法。

1、不冲撞性：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。

2、不间断性：每个机器人对图中每个结点无限地的访问。

这两个形式使用LTL表示如下：

不冲撞性主要满足同一时刻一个结点不能有两个机器人称为No\_collision，和相邻机器人不能朝对方移动称为No\_switch。

不间断性是确保每个机器人无限地访问图中每个结点，称为Live性质。

在访问过程中确保公平属性，表示为Liveness

### 移动算法的描述和定义

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算法如下：

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Legitimate phase | | | | | |
| RL1:: |  |  |  | → | r.Back |
| RL2:: |  |  |  | → | r.Front |
| RL3:: |  |  |  | → | r.Front |



图七，最小移动算法的稳定阶段

如图七所示，C1中机器人A快照顺时针快照 ，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。像这样的系统过程，称为稳定阶段。稳定阶段的系统状态称为稳定状态，系统进入稳定状态，就不会离开稳定阶段。

与稳定状态不同的是收敛阶段，系统除了稳定状态之外，还有其他的状态，其他任意状态可以通过收敛阶段，进入稳定状态。

最小移动算法中的收敛阶段的移动算法如下：

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Convergence phase | | | | | |
| RC1:: | 4 ≤ x ≤ z | ⋀ |  | → | r.Front |
| RC2:: | x ≠ y , x ˃ 0 | ⋀ |  | → | r.Doubt |
| RC3:: | 0 ˂ x ˂ z ˂ y ⋀ (x,y) ≠ (1,2) | ⋀ |  | → | r.Front |
| RC4:: |  |  |  | → | r.Back |
| RC5:: |  |  |  | → | r.Back |

# 基于NuXmv的移动算法的模型和验证

在完全同步调度策略(FSYNC,Fully Synchronous)、半同步调度策略(SSYNC,Semi Synchronous)、完全异步调度策略(ASYNC,Asynchronous)三种调度策略下，使用NuXmv分别对机器人移动算法进行建模，在建模的基础上，验证机器人移动算法是否满足永恒探索。本章节主要以建模、验证最小移动算法是否是永恒探索算法为范例，详述建模验、验证过程。

## NuXmv的简单介绍

NuXmv是一种符号模型检测工具，支持有限状态和无限状态系统的验证分析。在有限状态的情况下，是基于SAT算法实现。而对于无限状态的情况下，是基于SMT验证工具实现。

NuXmv关键字的功能很强大，例如init关键字可以给状态赋初始值，next可以关联前一个状态和后一个状态之间数量关系，同时支持条件选择，例如if-then-else、case...easc等。NuXmv的模块声明MODULE，构建模型更加直观。使用关键字LTLSPEC可以在NuXmv代码中直接使用LTL公式描述性质。支持交互式验证，容易获取状态转换路径。

## 快照F-R序列的计算

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。



图八，顺时针和逆时针快照获取

如图八中所示，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。顺时针时，b>a说明不包含位置为0的情况，b<a说明包含位置为0的情况。逆时针时，说明包含位置为0的情况，说明包含位置为0的情况。

其中，表示顺时针，表示逆时针。n是环形探索空间的位置结点的个数。利用上述公式，计算相邻机器人之间的无机器人的位置个数，进一步可知每个机器人视觉快照F-R序列。

## 机器人随机调度模型

对于每个机器人来讲，在调度过程中，只有选中和非选中两种情况。给机器人设置一个调度变量dispatcher，dispatcher可以有choose(选中)、steady(非选中)取值。

在NuXmv中，在声明变量时，指定变量的可取值范围，使用关键字VAR给变量的声明定义：

可以看出，式子中指出的可取值范围为。在使用时，不给赋初始值，系统初始状态值也是随机的，即 的初始值可以是也可以是。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，的取值进行定义:

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

## 机器人模型

所有机器人都执行相同的移动算法，机器人通过机器人位置信息，计算获取F-R快照序列。对于一个机器人R来说，其每次发生移动之后，其位置结点编号c(R)发生变化。后续机器人获取快照信息也会随着位置的变化而变化。那么，在机器人模型中，需要传入其他机器人的位置信息，设置统一的移动算法。机器人模型的核心就是移动算法的表示。机器人移动有前进和后退，而前进和后退是以顺时针和逆时针的机器人F-R快照作为参照物。机器人在编号序的位置结点上的移动，就是其所在位置编号值+1或者-1。

假设机器人r的F-R快照满足移动算法L，使用LTL公式表示如下：

定义移动算法L的决策的移动方向是，那么机器人r位置结点编号的移动量的定义如下：

环形拓扑结构空间，对于同一机器人r来在位置的顺时针和逆时针快照F-R序列，即机器人r的快照F-R序列是对称的。若时，规则L的无论是前进或者后退，此时依据移动量公式可知，即机器人的移动量是1或者 -1。当然，机器人不能匹配任何一个移动规则时，其。

机器人进入移动阶段，其下一个位置结点编号为。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，做出对应的机器人移动决策。

## 任意初始状态和永恒探索验证LTL公式

空间探索时，机器人都有一个初始位置。机器人移动永恒探索算法是机器人任意初始状态下，机器人移动探索是永恒探索的。由于在机器人移动过程中，满足不冲撞性，所以，机器人在环形拓扑结构空间上，顺时针顺序是不变的，即按照顺时针，机器人A在机器人B的前面，在随后的状态中，机器人A一直会在机器人B的前面。

验证代码可以直接使用LTL公式描述机器人移动算法状态可满足的性质，如下所示：

对于任意机器人，对于环形拓扑空间上的每个位置结点，都可以重复访问。

## 最小移动算法在三种调度策略下的结果

在完全同步调度策略(FSYNC,Fully Synchronous)、半同步调度策略(SSYNC,Semi Synchronous)，验证最小移动算法是满足对环形拓扑结构空间的永恒探索，在完全异步调度策略(ASYNC,Asynchronous)下，不满足对环形拓扑结构空间的永恒探索。

完全同步调度策略验证结果如下：

|  |  |  |
| --- | --- | --- |
| 位置结点数n | 验证结果 | 执行时间(min) |
| 10 | 满足 | 06:55:13 |
| 11 | 满足 | 54:50:19 |
| 13 | 未知 | 时间超过1小时 |

半同步调度策略验证结果如下：

|  |  |  |
| --- | --- | --- |
| 位置结点数n | 验证结果 | 执行时间(min) |
| 10 | 满足 | 01:27:10 |
| 11 | 满足 | 12:28:36 |
| 13 | 满足 |  |

完全异步调度策略验证结果如下：

|  |  |  |
| --- | --- | --- |
| 位置结点数n | 验证结果 | 执行时间() |
| 10 | 不满足 | 约5\*60 |
| 11 | 不满足 |  |
| 13 | 不满足 |  |

# 验证结果分析

使用NuSMV符号化模型检测方法对最小机器人移动算法，分别在完全异步模型、半同步模型、异步模型下建模，成功获得判定结果，此结果与其他文献的结论是一致的。最小机器人移动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，而在异步模型中不满足，根据实验结果给出的反例状态路径，可以得出其原因是异步过程中，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，机器人都不能移动，此时详尽过程与Ha Thi Thu Doan的文章中所获得反例原因是一样的，在他的文章中定义此种情况为死锁状态。

本文在Béatrice Bérard、Ha Thi Thu Doan取得的研究成果的基础上，实现了移动机器人符号化模型检测方法，并在Béatrice Bérard、Ha Thi Thu Doan的基础上有所进步，NuSMV中使用LTL描述机器人任意初始状态，验证机器人移动算法是否是永恒探索算法，这种验证方式更加完备，贴近永恒探索算法初始条件，在Ha Thi Thu Doan也提及了机器人位置初始转态，在他的文献中验证了固定某一初始状态下，实现移动算法的验证，对于每种初始状态，都手动定义，在位置结点数量和机器人数较少的时候，这种方式还比较容易操作，但是面对结点数量和机器人数量增多时，这种方法很容易遗漏初始状态，按照排列组合，每增加一个结点或者机器人数，初始状态可能是指数级的增长。使用NuSMV符号化模型检测方法在验证最小机器人移动算法时，模型为完全异步模型、半同步模型，若设定机器人数初始位置状态，电脑配置cpu intel E3-1231 v3 内存16G的情况下，一般计算时间15s以内。

# 总结以及未来工作

References:

1. Wang Q, Wu SJ, Li MS. Software defect prediction. Ruan Jian Xue Bao/Journal of Software, 2008,19(7):15651580 (in Chinese with English abstract). http://www.jos.org.cn/1000-9825/19/1565.htm
2. Hall T, Beecham S, Bowes D, Gray D, Counsell S. A systematic literature review on fault prediction performance in software engineering. IEEE Trans. on Software Engineering, 2012,38(6):12761304.
3. Yu SS, Zhou SG, Guan JH. Software engineering data mining: A survey. Journal of Frontiers of Computer Science and Technology, 2012,6(1):131 (in Chinese with English abstract).
4. Radjenovic D, Hericko M, Torkar R, Zivkovic A. Software fault prediction metrics: A systematic literature review. Information and Software Technology, 2013,55(8):13971418.
5. Akiyama F. An example of software system debugging. In: Proc. of the IFIP Congress. 1971. 353359.
6. Halstead MH. Elements of Software Science (Operating and Programming Systems Series). New York: Elsevier Science Inc., 1977.
7. McCabe TJ. A complexity measure. IEEE Trans. on Software Engineering, 1976,2(4):308320.
8. Chidamber SR, Kemerer CF. A metrics suite for object oriented design. IEEE Trans. on Software Engineering, 1994,20(6): 476493.
9. Basili VR, Briand LC, Melo WL. A validation of object-oriented design metrics as quality indicators. IEEE Trans. on Software Engineering, 1996,22(10):751761.
10. Subramanyam R, Krishnan MS. Empirical analysis of CK metrics for object-oriented design complexity: Implications for software defects. IEEE Trans. on Software Engineering, 2003,29(4):297310.
11. Zhou YM, Xu BW, Leung H. On the ability of complexity metrics to predict fault-prone classes in object-oriented systems. Journal of Systems and Software, 2010,83(4):660674.
12. Zhou YM, Leung H, Xu BW. Examining the potentially confounding effect of class size on the associations between object- oriented metrics and change-proneness. IEEE Trans. on Software Engineering, 2009,35(5):607623.
13. Zhou YM, Xu BW, Leung H, Chen L. An in-depth study of the potentially confounding effect of class size in fault prediction. ACM Trans. on Software Engineering and Methodology, 2014,23(1):10:110:51.
14. Zhao YY, Yang YB, Lu HM, Zhou YM, Song QB, Xu BW. An empirical analysis of package-modularization metrics: Implications for software fault-proneness. Information and Software Technology, 2015,57:186203.
15. Yang YB, Zhou YM, Lu HM, Chen L, Chen ZY, Xu BW, Leung H, Zhang ZY. Are slice-based cohesion metrics actually useful in effort-aware post-release fault-proneness prediction? an empirical study. IEEE Trans. on Software Engineering, 2015,41(4): 331357.
16. Sarkar S, Kak AC, Rama GM. Metrics for measuring the quality of modularization of large-scale object-oriented software. IEEE Trans. on Software Engineering, 2008,34(5):700720.
17. Meyers TM, Binkley D. An empirical study of slice-based cohesion and coupling metrics. ACM Trans. on Software Engineering and Methodology, 2007,17(1):2:127.

附中文参考文献:

[1] 王青,伍书剑,李明树.软件缺陷预测技术.软件学报,2008,19(7):15651580. http://www.jos.org.cn/1000-9825/19/1565.htm

[3] 郁抒思,周水庚,关佶红.软件工程数据挖掘研究进展.计算机科学与探索,2012,6(1):131.

1. \* 基金项目: 国家自然科学基金(00000000, 00000000); 南京大学计算机软件新技术国家重点实验室开放课题(KFKT00000000)

   Foundation item: National Natural Science Foundation of China (00000000, 00000000); State Key Laboratory for Novel Software Technology (Nanjing University)开放课题 (KFKT00000000)

   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 [↑](#footnote-ref-1)