ISSN (Print) : 0974-6846
Indian Journal of Science and Technology, *Vol 9(48), DOI: 10.17485/ijst/2016/v9i48/96047, December 2016* ISSN (Online) : 0974-5645

Temporal Modelling and Verification of Multi-Robot Concurrent Activities

Syed Asad Raza Kazmi*, Ayesha Naeem and Awais Qasim

Department of Computer Science, Government College University, Lahore, Pakistan;
arkazmi@gcu.edu.pk, ayesha2307@live.com, awais@gcu.edu.pk

Abstract

Objectives: In this work, we have introduced a method for formally verifying the concurrent activities of multiple robots. The objective is to formalize a framework for concurrent systems by using temporal logics. **Methods/Statistical Analysis:** The movement of the robots in given environment is modelled as a weighted transition system and the target is specified in Linear Temporal Logic (LTL) formulae over a set of propositions. **Findings:** The LTL formulas are characterized to handle the uncertainty or any abnormal activity during the process. The system is designed using SPIN model checker in which different LTL properties are verified. A theoretical description is supported by some experimental results, generated using the existing logics and techniques. **Application/ Improvement:** This work can be implemented for the surveillance task in pickup and delivery network.

Keywords: Formal Verification, Model Checking, Path Planning, Robotics

1. Introduction

Formal methods have been actively used to prove the correctness of safety critical systems. Different techniques of formal verification rely on the formalization of the system into a mathematical language. Several formal languages are available targeting different aspects of the critical systems that can formalize their design using one of these languages[1]. Recently the interest in using temporal logics has been increased for mission specification of robotic systems[2]. Temporal logics provide temporal operators that can be used to specify and verify various properties of interest for complex mission of these robotic systems.

In classical research of robots path planning problem, the aim is to control the movement of robots from the start location to the desired destination location, while eluding any obstacle during this process[3]. In current research, the finite systems and motion planning by using LTL specification is more concerned[4]. Temporal logics provide different dialects for specifying complex mission for robots[5,6]. The power of these dialects lies in the tools of model checking[7,8,9] that can generate the paths while satisfying the common mission specifications. Industrial applications are concentrating on the use of formal methods at the earlier stages of design to prove their correctness[10]. The proposed method is used to reduce the complexity of the system by satisfying the optimization propositions. Different processes depend on the trace-closeness concept that was applied in multiple robot motion planning[11] and solve these problems by formal techniques[12]. These requirements of satisfying the optimization propositions for the mobile robot which is a part of team of robots are even more challenging[13]. Different formal verification techniques like model checkers are powerful mathematical tools[14] for the modelling and analysis of manufacturing systems, hardware structures, real time systems, health and medical systems, defence systems, railway networks, protocols and networks, operations research and performance evaluation. The use of model checking to ensure the correctness of real-time systems is highly desirable[15,16]. When a robot moves in

* *Author for correspondence*

a changing environment it needs to be flexible for such motion changing primitives[17], hence multi robot coverage and task allocation are fundamental problems in robotic systems. Work in the outdoor robotics has been concerned with maintaining the safety of robot while moving in the dynamic environment[18]. Multi robot tasks have been limited to path tracking while avoiding the obstacles along the way[19]. Formal modelling is recommended for such critical systems and verification of their activities with temporal logic constraints is desirable. Temporal logics are very effective for the systems in which reactive mission and motion planning is more considerable. Robots can follow a given trajectory and if any abnormal activity occurs, then such cases are handled by generating the transition systems according to the situation.

In this research, first we provide an algorithm to construct a transition system that helps to capture the concurrent moves of a group of robots. Then an algorithm is given that generate accurate run guaranteeing their correctness. Finally a model is presented that generate the moves for a group of robots satisfying general LTL formulas. A formalized framework is discussed to overcome the problems in which robots lack accuracy in the presence of uncertain activities.

The rest of the paper is organized as follows. In section 2 we primarily discuss the modelling of concurrent activities of the robots. In section 3 we construct the transition system and generate optimal run by running the system automaton. We present the LTL properties and implementation results in Section 4. Finally section 5 concludes the paper.

2. Problem Formulation

We consider the case where robotic teams have to accomplish a mission and achieve the given target. Π represents finite set of atomic propositions. We assumed an optimizing proposition seΠ corresponding to particular task that is repeated infinitely often and situation is represented with LTL formula:

$$\varphi := \psi \wedge GF\sigma \quad (1)$$

Here φ represents mission specification, the LTL formula over Π is ψ and GFσ represents that the proposition σ is satisfied repeatedly. We specify the behavior of robots in the form of infinite sequence as $T = \{T(1), T(2),\}$. Here $T(x)$ is the instance of time when σ is satisfied for the time by the robots. The cost function is defined as;

$$C(T) = \lim_{x^{\circledR}+¥} (T(x+1)-T(x)) \quad (2)$$

Figure 1 shows the complete working of the proposed algorithm.

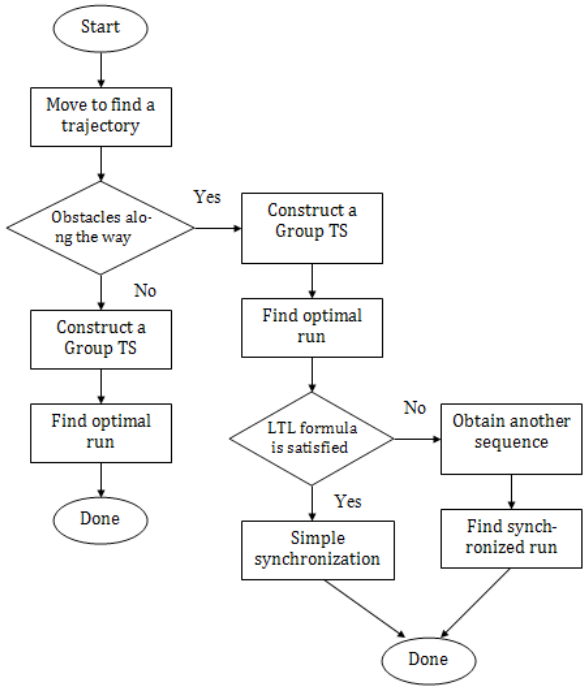

Figure 1. An overview of the working of the proposed algorithm.

3. Obtaining Transition System for Concurrent Moves

We capture the behavior of team members while moving in an environment. We define a way to obtain the position whether the robots are at the region or traveling between the regions. There are different approaches for the verification of concurrent systems by model checking20. To formally model the moves in the environment, we use a transition system $T=(Q_T, {}^{0}_{q_T}, d_T, P_T, L_T, w_T)$, where Q_T is set of states, ${}^{0}_{q_T}$ represents starting state, dT represents the set of transitions, PT is set of propositions, LT is mapping on Q_T and w_T is transition weight. Now we construct the T by running DFS on the transition system of each member of the group.

3.1 Optimized Solution

In this section the system for the team of robots in a specific environment is elaborated. Given the individual systems of robots as {,}, Algorithm 1 is used to construct the group transition system T that captures all the serialized moves of the robots. The joint behavior of robots is presented in this section in the form of an automaton and then optimal run table is generated that shows the results after running the automaton.

Algorithm 1: Group-TS
Input: Transition system of each member {T1, T2, T3}.
Output: Combined Transition System of members that is **T**.

- Let m = 1, 2, 3
- $=. {}^{0}_{q_T} = {}^{0}_{q_M}$
- Recursive search on **T** starting from initial state (${}^{0}_{q_T}$).
- $q_m \in q$.
- The transition of T_m is defined as $\rightarrow_m$, where $\rightarrow_m$ = (q,q')
- τ : set of possible transitions and $\rightarrow_m \in \tau$ then do.
- w = minimum weight of $\rightarrow_m$ when the robot finds some region.
- Find new state of transition system that is q' and if q' not exists then.
- Insert the state q' to **T**.
- Insert new transition to μιν δ_T with assigning the weight w. where δ_T is the set of transitions.
- Prolong search from new state q'
- else if there is no transition for (q,q') then
- Include transition that is from (q,q') to δ_T with assigning the weight w.

In the Algorithm 1 a recursive depth first search is applied that begins at initial state of group transition system T. We define the as the tuple of initial states of m transition systems. When all states and transitions have been discovered then this algorithm will stop it's searching for the transition system T.

Figure 2, Figure 3 and Figure 4 represents the automaton, and respectively as the transition systems for the three robots that are in a network with four vertices. These states are represented as {,,,} and their motion capabilities are shown by edges. The weights represent traveling time between the two states. The transition system **T** as depicted in Figure 5, shows the group automaton that captures the behavior of three robots combined in 6 states. The state (,,) means that the first robot is at location , second robot is at and third robot is at region . By running the transition system T and the formula φ := ψ∧GFσ given in the above automaton results in an optimal run.

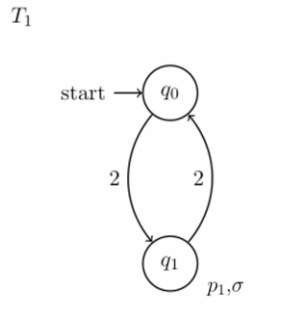

Figure 2. Transition system for robot1.

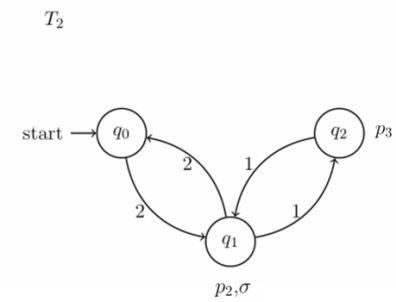

Figure 3. Transition system for robot2.

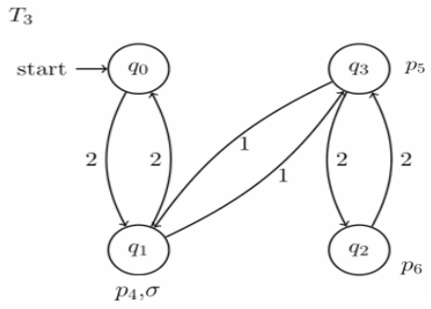

Figure 4. Transition system for robot3.

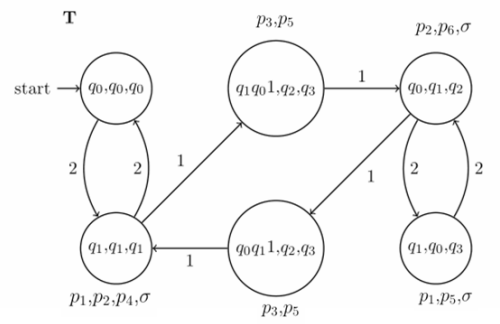

Figure 5. Combined transition system for the 3 robots.

In Table 1 the first row shows the time when the transition occurred, second row represents the run , third row corresponds to the satisfied propositions, and last three rows shows the separate run of these three robots. We observed in the optimal run that (,,), (,,), (1,,) is prefix cycle and (,,), (,,) is a suffix cycle. These cycles will be repeated an infinite number of times. The given time which satisfies the σ is = {2,4,6,8,10,…} and the function defined in (2) is $C(T) = 2$. The time sequence when σ is repeatedly satisfied is = {2,4,6,8,10,…} and cost function is given as;

$$C(T) = (T(x+1) - T(x)) \quad (3)$$

$$= T(x+1) - T(x) \quad (4)$$

$C(T) = 2$ is the obtained function, where σ is repeatedly satisfied. Similarly Table 2 shows another sequence for different runs. In Algorithm 2 the exact solution is summarized and it shows that a particular solution is given to the specified problem for that case where robots have the exact time information.

4. Implementation and Results

We considered the scenario of a robotic environment where robots have to pick some data and drop it at the required position. Transition systems, and model the movements of robots and their combined moves are established in T showing the concurrent behavior of the robots.

For the experiment, the transition systems, and are identical excluding their initial states. Now the atomic propositions for data pick and data drop performed at some regions by robots are defined as:

$$\prod = \{pick, drop, R_1pick, R_2pick, R_3pick, R_1drop, R_2drop, R_3drop, pick_1, pick_2, drop_1, drop_2, R_1pick_1, R_1pick_2, R_2pick_1, R_2pick_2, R_3pick_1, R_3pick_2, R_1drop_1, R_1drop_2, R_2 drop_1, R2drop_2, R3drop_1, R3drop_2\} \quad (5)$$

The propositions *pick* and *drop* means that data has been picked and dropped, respectively. The propositions of the form pickX and dropX, where X∈ {1,2} are more expressive and captures the data

pick and data drop locations as well. *Pick 2* shows that data has been picked at pick location 2. Propositions that are in the form pick and drop, where Y∈ {1,2}, represents that robot Y has picked and dropped the data, respectively.

We considered different scenarios and then specified the problem as LTL formulas. The mission specification example for a case as "Group of robots picks the data in simultaneous way and delivers the data before picking it again". This mission problem can be specified in the form of LTL formula as:

$$\varphi = G(pick \Rightarrow X(\emptyset pick\ U drop)) \wedge G(pick \Rightarrow X(\emptyset pick\ U drop)) \wedge G(pick \Rightarrow X(\emptyset pick\ U drop)) \wedge GF\sigma \quad (6)$$

Here σ = pick is a satisfying proposition. The syntactically modified formula for SPIN model checker is given as:

Table 1. Results after running the transition system

T	0	2	3	4	6	8	...
r^*_{group}	q_0,q_0,q_0	q_1,q_1,q_1	$q_1q_0^1,q_2,q_3$	q_0,q_1,q_2	q_1,q_0,q_3	q_0,q_1,q_2	...
L_T	.	$p_1,p_2,p_4,\square$	p_3,p_5	$p_2,p_6,\square$	$p_1,p_5,\square$	$p_2,p_6,\square$	...
r_1^*	q_0	q_1	.	q_0	q_1	q_0	...
r_2^*	q_0	q_1	q_2	q_1	q_0	q_1	...
r_3^*	q_0	q_1	q_3	q_1	q_0	q_1	...

Table 2. Results after running the transition system

T	0	2	3	4	5	6	7	…
r^*_{group}	q_0,q_0,q_0	q_1,q_1,q_1	$q_1q_0^1,q_2,q_3$	q_0,q_1,q_2	$q_0q_1^1,q_2,q_3$	q_1,q_1,q_1	$q_1q_0^1,q_2,q_3$	…
$L_T(.)$	.	$p_1,p_2,p_4,\square$	p_3,p_5	$p_2,p_6,\square$	p_3,p_5	$p_1,p_2,p_4,\square$	p_3,p_5	…
r_1^*	q_0	q_1	.	q_0	.	q_1	.	…
r_2^*	q_0	q_1	q_2	q_1	q_2	q_1	q_2	…
r_3^*	q_0	q_1	q_3	q_1	q_3	q_1	q_3	…

[ ] ((r1p) -> X (! (r1p) U (r1d))) && [ ] ((r2p) -> X (! (r2p) U (r2d))) && [ ] ((r3p) -> X (! (r3p) U (r3d))) && [ ] <> (a) (7)

4.1 System Modeling Using SPIN Model Checker

The work is implemented in Promela for the analyses of models of different systems with SPIN model checker[9]. Promela is a process meta-language[21] that is used to verify the correctness of the models. Different processes are created for each robot and then the grouped transition system is modeled in SPIN. An information message is generated for path confirmation. LTL formulas are also implemented in SPIN and the resulted automatons for each formula are shown. In Figure 6, the automata of path confirmation of multi robots are shown. When a robot starts its move, it first confirms the path availability then it moves. If the move is valid then an acknowledgement message is passed. In case a move is invalid then a failed message is passed.

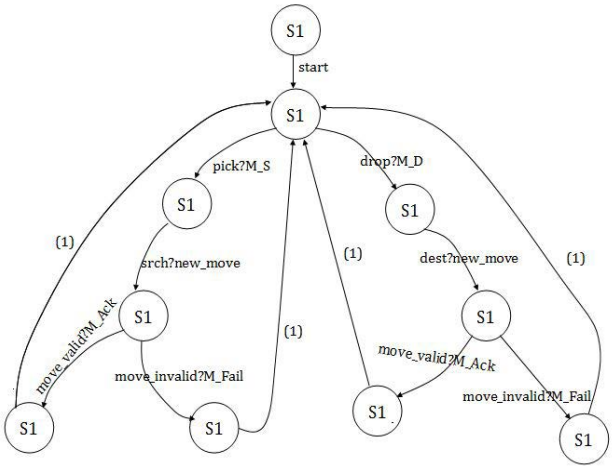

Figure 6. Automata for path confirmation of multi robots.

4.2 Verification of LTL Formulas

In this section we specify the properties of interest as LTL formulas for their verification using SPIN model checker. SPIN converts the LTL formulae to Buchi automata and Buchi automaton accepts set of infinite traces.

- The first case is that each robot visit data picking location to pick the data until they drop the data at delivering location. This mission can be expressed in the form of LTL formula as given below.

$$\varphi = G(pick \Rightarrow X(\emptyset pick\ U\ drop) \wedge GF\sigma \quad (8)$$

The above formula in spin can be written as:

$$[\ ](p \text{ -> } X(\ !\ (p)\ U\ (d))\ ) \&\&\ [\ ] <> a \quad (9)$$

- In another case robots must pick their data and then start their move.
-

$$\varphi = G(pick \Rightarrow (pick\ \grave{U} pick\ \grave{U} pick)) \wedge GF\sigma \quad (10)$$

The above formula in SPIN can be written as:

$$[\ ]\ (\ p \text{ -> } (\ (r1p)\ \&\&\ (r2p)\ \&\&\ (r3p)\ ))\ \&\&\ [\ ] <> a \quad (11)$$

We verified the different properties of interest in SPIN. Table 3 shows the results of syntax check, redundancy check, process execution, verification of safety and liveness properties.

Table 3. Results after compilation

Property	Property Verification Result
Syntax check	Model contains 4 claims and these claims are used in verification run, there's no syntax error.
Redundancy check	2 redundancies are detected that are removed to reduce the complexity.
Execution of processes	8 processes are created successfully and a queue is established for this process execution.
Safety property	Full state space search is done for processes and no errors found in it.
Liveness property	Same results as for safety property and for depth search no errors found.

5. Conclusion

In this work we formally modeled the concurrent activities of a group of robots using temporal logics. The specifications were expressed as LTL formulae and an algorithm was provided to model the transition system for a group of robots. Our method is optimal in a computational way as compared to the previous methods. The main drawback of previous work is that their models are very computationally expensive and becomes difficult to implement. For the extension of this work the constraints can be categorized into environment-related constraints, robot-related constraints and task-related constraints and an effective way is to build a hierarchical mechanism.

6. References

1. Chiappini A, Cimatti A, Porzia C, Rotondo G, Sebastiani R, Traverso P, Villafiorita A. Formal specification and development of a safety-critical train management system. In International Conference on Computer Safety, Reliability, and Security Springer Berlin Heidelberg, Lecture Notes in Computer Science. 1999 Oct 14; 1698:410–9.
2. Wongpiromsarn T, Topcu U, Murray RM. Receding horizon control for temporal logic specifications. In Proceedings of the 13th Association for Computing Machinery (ACM) International Conference on Hybrid Systems: Computation And Control, USA; 2010 Apr. p. 101–10.
3. Rekleitis I, New AP, Rankin ES, Choset H. Efficient boustrophedon multi-robot coverage: an algorithmic approach. Annals of Mathematics and Artificial Intelligence. 2008 Apr; 52(2–4):109–42.
4. Tabuada P, Pappas GJ. Model checking LTL over controllable linear systems is decidable. In International Workshop on Hybrid Systems: Computation and Control Springer Berlin Heidelberg, Lecture Notes in Computer Science. 2003 Mar 14; 2623:498–513.
5. Bhatia A, Kavraki LE, Vardi MY. Sampling-based motion planning with temporal goals. In the Proceedings of Institute of Electrical and Electronics Engineers (IEEE) International Conference on Robotics and Automation (ICRA), Anchorage, Alaska, USA; 2010 May 3–7. p. 2689–96.
6. Kress-Gazit H, Fainekos GE, Pappas GJ. Temporal-logic-based reactive mission and motion planning. Institute of Electrical and Electronics Engineers (IEEE) transactions on robotics. 2009 Dec; 25(6):1370–81.
7. Calzolai F, Nicola DR, Loreti M, Tiezzi F. TAPAs: a tool for the analysis of process algebras. In Transactions on Petri Nets and Other Models of Concurrency I Springer Berlin Heidelberg, Lecture Notes in Computer Science. 2008; 5100:54–70.
8. Cimatti A, Clarke E, Giunchiglia E, Giunchiglia F, Pistore M, Roveri M, Sebastiani R, Tacchella A. NuSMV 2: An open source tool for symbolic model checking. In International Conference on Computer Aided Verification Springer Berlin Heidelberg, Lecture Notes in Computer Science. 2002 Sep; 2404:359–64.
9. Holzmann GJ. The model checker SPIN. Institute of Electrical and Electronics Engineers (IEEE) Transactions on software engineering. 1997 May; 23(5):279–95.
10. Woodcock J, Larsen PG, Bicarregui J, Fitzgerald J. Formal methods: practice and experience. Association for Computing Machinery (ACM) Computing Surveys (CSUR). 2009 Oct; 41(4):19.
11. Chen Y, Ding XC, Stefanescu A, Belta C. Formal approach to the deployment of distributed robotic teams. Institute of Electrical and Electronics Engineers (IEEE) Transactions on Robotics. 2012 Feb; 28(1):158–71.
12. Yarmohamadi M, Javadi HH, Erfani H. Improvement of robot path planning using particle swarm optimization in dynamic environments with mobile obstacles and target. Advanced Studies in Biology. 2011; 3(1):43–53.
13. Kim JH, Vadakkepat P. Multi-agent systems: a survey from the robot-soccer perspective. Intelligent Automation and Soft Computing. 2000 Jan; 6(1):3–17.
14. Frappier M, Fraikin B ,Chossart R , Fa RCY , Ouenzar M . Comparison of model checking tools for information systems. In 12th International Conference on Formal Engineering Methods (ICFEM), Lecture Notes in Computer Science, Springer. 2010 Nov; 6447:581–96.
15. Qasim A, Kazmi SA, Fakhir I. Executable semantics for the formal specification and verification of E-agents. Indian Journal of Science and Technology. 2015 Jul; 8(16):1–8.
16. Qasim A, Kazmi AR, Fakhir I. Formal specification and verification of real-time multi-agent system using timed arc Petri nets. Advances in Electrical and Computer Engineering. 2015 Jan; 15(3):73–8.
17. Navarro I, Matía F. A survey of collective movement of mobile robots. International Journal of Advanced Robotic Systems. 2013 Jan; 10(73):1–9.
18. Clark CM, Rock SM, Latombe JC. Motion planning for multiple mobile robots using dynamic networks. In the Proceedings of Institute of Electrical and Electronics Engineers (IEEE) International Conference on Robotics and Automation (ICRA),USA. 2003 Sep; 3:4222–7.
19. Guo Y, Parker LE. A distributed and optimal motion planning approach for multiple mobile robots. In the Proceedings of Institute of Electrical and Electronics Engineers (IEEE) International Conference on Robotics and Automation (ICRA), USA. 2002 May; 3:2612–9 .
20. Vardi MY , Wolper P . An automata-theoretic approach to automatic program verification. In 1st Symposium in Logic in Computer Science (LICS). Institute of Electrical and Electronics Engineers (IEEE) Computer Society, Cambridge; 1986. p. 322–31.
21. Jiang K. Model checking C programs by translating C to promela [Master thesis]. Sweden, Institutionen för informationsteknologi, Uppsala Universitet; 2009. p. 1–81.