Modeling Autonomous Vehicles at an Intersection without a Traffic Light

Yui Sahara* Mazaki Nakamura* *Nakamura Lab.

Abstract

Automatic driving technology has been in rapid development. For safe and efficient cities with a lot of such autonomous vehicles, we need to consider not only control systems for individual vehicles but also those for a group of vehicles. In this article, we investigate a way to model and verify of autonomous-vehicle at intersection without traffic light using a model checking technique.

1 Introduction

Automatic driving technology is in rapid development. Automatic driving is divided into five levels depending on the technology installed. In Japan, general cars install level 2 supporting driver. In the future, Japan government will set a goal that the vehicles installed level 4 become popular. If a lot of automatic driving cars are used in a city, some problems may occur. Thus, we need autonomous-vehicle group control algorithms. In this paper, we verify the autonomous-vehicle group control algorithms with formal method.

2 Past research

Autonomous-vehicle systems support drivers to avoid conflict and to control safety. Cars are necessary for our lifestyles, but traffic accidents happen anytime. Therefore, we ought not drive car. Autonomous-vehicle is necessary for human life in the future. A collision avoidance systems were developed to protect drivers.

3 Methods

3.1 Model checking tool UPPAAL

UPPAAL [1] is an integrated tool environment for modeling, simulation, and verification of real-time systems. UPPAAL provides graphical interface to describe timed automata and simulate and verify them. UPPAAL's timed automata is

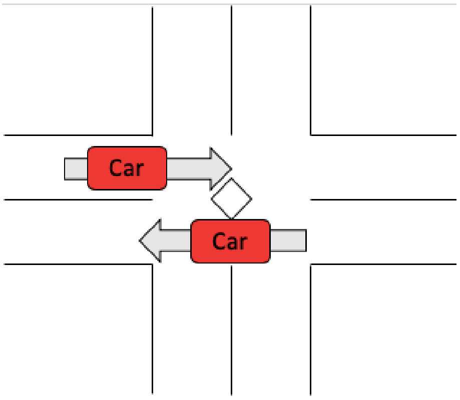


Figure 1: Model of Intersection without Traffic Lights

an extension of usual timed automata. We declare local and global variables for data types and clocks. In UPPAAL, a system model consists of collection of timed automata. Global variables are declared for a system model. Timed automata [2] consist of a finite number of clock variables, a finite number of locations, and transitions between locations. A clock variable stores time progress of continuous state of the system.

3.2 Model of Intersection

An intersection without a traffic light is one lane on one side(Fig.1). Cars that turn left can pass at the same time in this intersection.

4 Results

4.1 Timed Automaton of A Intersection

In this section we give a system model of intersections with several autonomous vehicles. There is a timed automata of intersection without traffic lights using UPPAAL(Fig.2). This is template of timed automata. Parameters are declared direction of cars, cars' start points, and cars' end points. Global declarations and local declarations are shown Fig.3 and Fig.4. Moreover, we declare systems instance(Fig.5). System declarations make timed automata from template.

4.2 Verification

A car takes from seven units time to seventeen units time to pass the intersection. In this section, we verify the shortest time of four cars' passing an intersection. Verification uses like next formula. First formula verify reachability, second formula verify minimality. Verification results that shortest time is 14 units

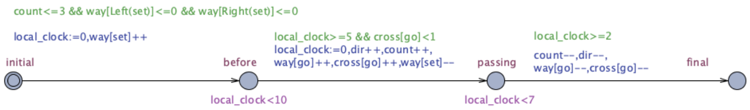
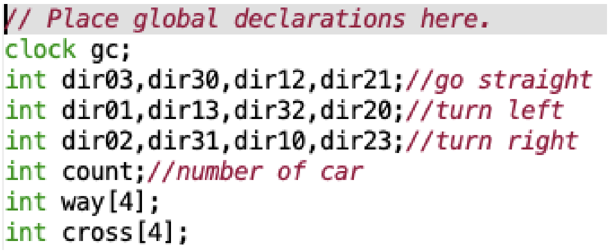

count<=3 && way[Left(set)]<=0 && way[Right(set)]<=0
local_clock:=0,way[set]++
initial
before
local_clock<10
local_clock>=5 && cross[go]<1
local_clock:=0,dir++,count++,
way[go]++,cross[go]++,way[set]--
passing
local_clock<7
local_clock>=2
count--,dir--,
way[go]--,cross[go]--
final

Figure 2: Template of Timed Automata



```
// Place global declarations here.
clock gc;
int dir03,dir30,dir12,dir21;//go straight
int dir01,dir13,dir32,dir20;//turn left
int dir02,dir31,dir10,dir23;//turn right
int count;//number of car
int way[4];
int cross[4];
```

Figure 3: Global declarations

time.

$$E <> (gc == 14andns.finalandsn.finalandew.finalandwe.final)$$

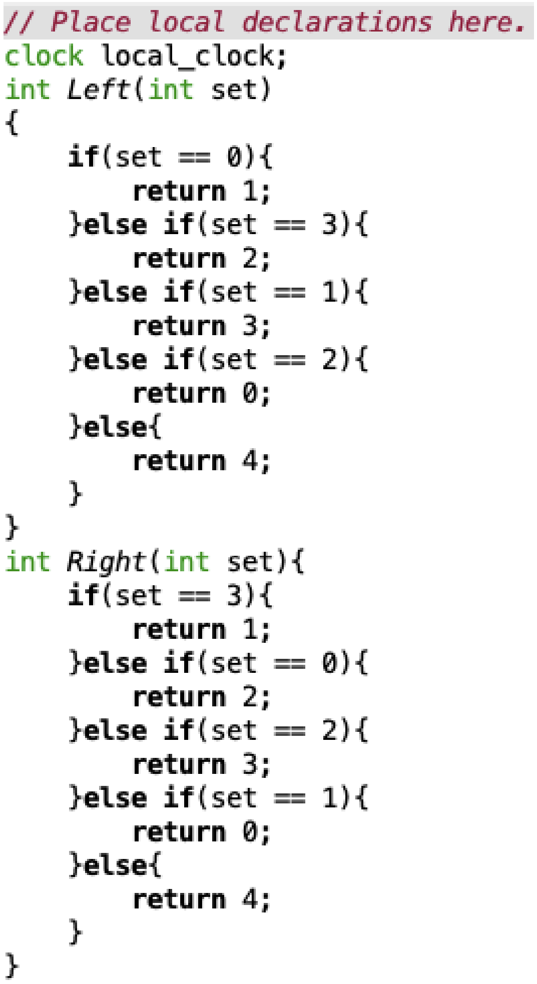
$$A[](gc < 14implynot(ns.finalandsn.finalandew.finalandwe.final))$$

5 Conclusion

We proposed a modeling and verification method of autonomous-vehicle group control algorithms using UPPAAL. We could verify and modeling the behavior of autonomous vehicles at an intersection.

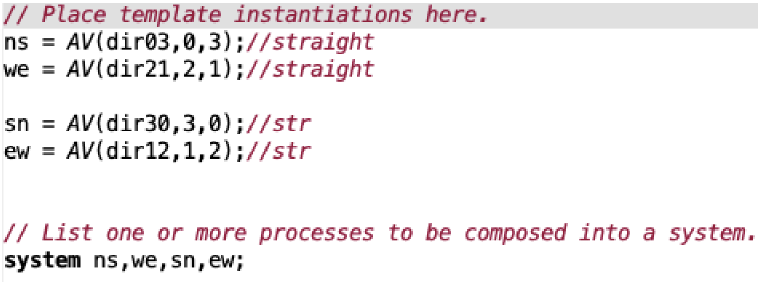
Biographical Statement

Yui Sahara is graduate student in Nakamura Laboratory at Toyama Prefectural University in the Graduate School Engineering majoring Information Systems Engineering. She received Bachelor of Engineering from Toyama Prefectural University at 2019. Her recent work includes ”Modeling and verification of autonomous-vehicle group control algorithms using UPPAAL”(ISCIE, May, 2019). She is researching performance model verification of autonomous-vehicle group control algorithms using a model checking technique.



```
// Place local declarations here.
clock local_clock;
int Left(int set)
{
    if(set == 0){
        return 1;
    }else if(set == 3){
        return 2;
    }else if(set == 1){
        return 3;
    }else if(set == 2){
        return 0;
    }else{
        return 4;
    }
}
int Right(int set){
    if(set == 3){
        return 1;
    }else if(set == 0){
        return 2;
    }else if(set == 2){
        return 3;
    }else if(set == 1){
        return 0;
    }else{
        return 4;
    }
}
```

Figure 4: Local declarations



```
// Place template instantiations here.
ns = AV(dir03,0,3);//straight
we = AV(dir21,2,1);//straight

sn = AV(dir30,3,0);//str
ew = AV(dir12,1,2);//str


// List one or more processes to be composed into a system.
system ns,we,sn,ew;
```

Figure 5: System declarations

References

[1] Kim Guldstrand Larsen and Paul Pettersson and Wang Yi, UPPAAL in a Nutshell, International Jour- nal of Software Tools for Technology Transfer, Vol.1, No.1-2, pp.134-152, 1997.

[2] Timed Automata: Semantics, Algorithms and Tools, Johan Bengtsson and Wang Yi. In Lecture Notes on Concurrency and Petri Nets. W. Reisig and G. Rozenberg (eds.), LNCS 3098, Springer-Verlag, 2004.