Supplementary Materials for “Safety Analysis of Autonomous Driving Systems: A Simulation-based Runtime Verification Approach”

Duong Dinh Tran, Takashi Tomita, and Toshiaki Aoki
Japan Advanced Institute of Science and Technology, Ishikawa 923-1292, Japan

***Abstract*—This document provides the language grammar of AWSIM-Script and the experimental results of the second run of the cut-out and deceleration scenarios.**

***Index Terms*—autonomous driving systems, safety verification, Autoware, safety standard, model checking**

I. AWSIM-Script language grammar

We use the ANother Tool for Language Recognition (ANTLR) v4 framework [1] to define the grammar of AWSIM-Script and to implement the parser component. This is a powerful parser generator framework, widely used to build interpreters, compilers, and tools that require parsing formal languages such as programming languages or domain-specific languages. ANTLR supports the grammar that is close to Backus-Naur Form (BNF) but includes additional features that are not present in standard BNF.

Listing 1 shows part of the language grammar defined in ANTLR. The language supports constructs for defining positions, road elements (e.g., lane changes), and simulation configurations. For example, a `positionExp` denotes a location on a traffic lane, and a `roadExp` allows users to define routes and lane changes together with desired speeds.

Note that `stringExp`, `numberExp`, and `idExp` in the grammar denote strings (which are enclosed by a pair of double quotes), float numbers, and variable/function names (which are made from alphabets, numbers 0-9, and underscore character), respectively, although we omit their definitions in the figure. We use the ANTLR parser generator tool to generate the base parser from the defined grammar. Building on this base parser, we implement the module that parses the input script and invokes the appropriate simulation APIs of the extended AWSIM-Labs to perform the simulation. Note also that there exists the case in which an input script is accepted by the grammar but the parsing process produces an error, for example, using an undefined variable.

II. Reliability of Experimental Results

Although it is inherently impossible to reproduce identical traces across experimental runs, our experimental results presented in the paper are reliable. To demonstrate this reliability, we conducted a second set of experiments for the cut-out and deceleration scenarios, with results presented in Tables I and II (the cut-in scenario was skipped due to the time and effort required).

Listing 1. Part of AWSIM-Script grammar defined in ANTLR v4.

```
grammar AWSIMScriptGrammar;
positionExp
  : stringExp (’at’ (numberExp | variableExp))?
  | variableExp (’back’ | ’forward’ | ’left’ | ’right’)
     (numberExp | variableExp)
  | positionExp (’back’ | ’forward’ | ’left’ | ’right’)
     (numberExp | variableExp);
roadExp
  : stringExp (’max-velocity’ ’(’ (numberExp | variableExp) ’)’)?
  | (’change-lane’ | ’cut-in’ | ’cut-out’) ’(’ argumentList? ’)’;
vector2Exp : ((numberExp | variableExp) ’#’
           (numberExp | variableExp));
configExp
  : ’aggressive-driving’
  | (’acceleration’ | ’deceleration’ | ’speed’) ’(’ (numberExp |
     variableExp) ’)’
  | (’delay-spawn’ | ’delay-move’) ’(’ (numberExp | variableExp)
     ’)’
  | (’delay-spawn-until-ego-move’ | ’delay-move-until-ego-move’)
     ’(’ (numberExp | variableExp) ’)’
  | (’delay-spawn-until-ego-engaged’ |
     ’delay-move-until-ego-engaged’) ’(’ (numberExp |
     variableExp) ’)’;
egoSettingExp : ’max-velocity’ ’(’ (numberExp | variableExp) ’)’;
simulationSettingExp : ’saving-timeout’ ’(’ numberExp ’)’;
functionExp : idExp ’(’ argumentList? ’)’ ;
arrayExp : ’[’ argumentList? ’]’;
argumentList : expression ( ’,’ expression )∗ ;
assignmentStm : variableExp ’=’ expression;
variableExp: idExp;
expression
  : stringExp | numberExp | vector2Exp | positionExp | roadExp |
     arrayExp | variableExp | configExp | egoSettingExp |
     simulationSettingExp | functionExp;
statement : (assignmentStm | functionExp) ’;’ ;
scenario : (statement)+ EOF;
```

The key results—collision and non-collision—were consistently preserved across the two runs. For non-collision cases, the minimum TTC values from the second runs were generally close to those from the first runs.

Factors contributing to the non-determinism. The first factor contributing to the non-determinism of the experimental results originates from the Autoware side. As shown in the two tables, the ego vehicle’s speeds at the moment when the cut-out and deceleration starts (**Actual Ve**) occasionally differed between the two runs.

Another contributing factor to the non-determinism of the traces was the behavior of the `Update` function in the Unity-based AWSIM-Labs source code. Unlike the deterministic `FixedUpdate` function, the timing and frequency of the `Update`

This work was supported by JST, CREST Grant Number JPMJCR23M1.

TABLE I
SECOND RUN OF CUT-OUT EXPERIMENTS.

Mode	ID	Config Ve, Vo	First Run			Second Run		
			Actual Ve	Collision	Min TTC	Actual Ve	Collision	Min TTC
Lidar	20-1	5.56	5.55	N	2.61	5.55	N	2.59
	20-2	5.56	5.56	N	2.74	5.55	N	2.74
	30-1	8.33	8.31	N	1.59	8.35	N	1.47
	30-2	8.33	8.34	N	1.38	8.28	N	1.37
	40-1	11.11	11.12	N	1.89	11.11	N	1.84
	40-2	11.11	11.12	N	1.88	11.12	N	1.90
	50-1	13.89	13.71	Y		13.72	Y	
	50-2	13.89	13.71	Y		13.63	Y	
Camera -lidar fusion	20-1	5.56	5.56	N	2.19	5.55	N	2.39
	20-2	5.56	5.55	N	2.48	5.55	N	2.29
	30-1	8.33	8.32	N	1.25	8.35	N	1.47
	30-2	8.33	8.29	N	2.09	8.34	N	1.39
	40-1	11.11	11.12	N	1.08	11.11	N	1.73
	40-2	11.11	11.12	N	1.41	11.12	N	1.42
	50-1	13.89	13.72	Y		13.72	Y	
	50-2	13.89	13.71	Y		13.67	Y	

TABLE II
SECOND RUN OF DECELERATION EXPERIMENTS.

Mode	ID	Config Ve, Vo	First Run			Second Run		
			Actual Ve	Collision	Min TTC	Actual Ve	Collision	Min TTC
Lidar	20	5.56	5.55	N	1.54	5.55	N	1.70
	30	8.33	8.33	N	1.59	8.34	N	1.57
	40	11.11	11.11	Y		11.08	Y	
	50	13.89	13.77	Y		13.75	Y	
Camera -lidar fusion	20	5.56	5.55	Y		5.55	Y	
	30	8.33	8.33	N	1.25	8.34	N	1.28
	40	11.11	11.12	Y		11.12	Y	
	50	13.89	13.76	Y		13.79	Y	

calls depend on the computing performance during execution[1]. This variability can result in non-deterministic outcomes due to inconsistencies between runs.

In the cut-out experiments, larger deviations in the minimum TTC values were observed in the camera-lidar fusion perception mode. This is likely due to the additional non-determinism introduced by the fusion and camera-based object detection systems.

REFERENCES

[1] T. Parr, *The Definitive ANTLR 4 Reference*. 2nd ed., 2013.

[1] https://docs.unity3d.com/6000.0/Documentation/ScriptReference/MonoBehaviour.html