Title	CoqによるBBSLの形式化と検証
Author(s)	宇田，拓馬
Citation	
Issue Date	2021-03
Type	Thesis or Dissertation
Text version	author
URL	http://hdl.handle.net/10119/17083
Rights	
Description	Supervisor:青木 利晃，先端科学技術研究科，修士（情報科学）

Formalization and Verification for BBSL with Coq

1910031 Takuma Uda

In recent years, technological development aimed at the practical application of autonomous driving has been actively carried out. Safety evaluation has become an important issue in autonomous driving systems. In addition, it is difficult to define safety requirements because it is large-scale and complicated, and various driving environments are assumed. Therefore, research is being conducted by the ministries and agencies of each country to define safety standards by systematizing autonomous vehicles and their surrounding environments. However, in those studies, the specifications are described using figures and natural language, which causes ambiguity in the content. Therefore, the meaning of the specification is not uniquely determined, and it is difficult to verify the safety. Formal specification is a method for describing specifications strictly without ambiguity. Since the meaning of the specification is strictly defined in the formal specification, its unambiguity can be guaranteed. However, in general formal specification languages such as Z and VDM, abstract description using figures and natural language is difficult. Therefore, the Bounding Box Specification Language (BBSL) has been proposed by previous research as a formal specification language for images in autonomous driving systems. BBSL is an original extension of the interval arithmetic system, and the focus is on formally describing the positional relationship of objects on an image using a Bounding box represented by a two-dimensional interval.

The purpose of this study is to improve the quality of the specifications described in BBSL by formalizing BBSL using Coq and verifying its language specifications. Since high safety is required for autonomous driving systems, high quality is required for BBSL, which is the language that describes the specifications. By formalizing BBSL, it becomes possible to describe reliable specifications. Coq, a theorem proving support system, is used to operate BBSL on a computer.

A language can be formalized by giving formal semantics to the syntax of the language. In this study, Coq is used to formalize BBSL. In other words, it is necessary to express BBSL on Coq. There are shallow embedding and deep embedding as methods for implementing a language on another language. In shallow embedding, the target language is evaluated by the semantics of the implementation language. Implementation is easy as an advantage, but the target language may not be implemented due to the limitation of the semantics of the implementation language. In deep embedding, evaluation is performed by implementing semantics in the abstract syntax of the target language. The advantage is that the semantics of the target language can be freely given, but the disadvantage is that the implementation becomes complicated. In this study, deep embedding is adopted to give BBSL a formal semantics. To formalize BBSL with deep embedding using Coq, first define the abstract syntax of BBSL. Second, we define a semantic function that gives semantics by associating mathematical objects with abstract syntax. In addition, since BBSL extends the interval system independently, it is necessary to define mathematical objects as well. Third, implement these on Coq.

Evaluation experiments will be conducted on the formalized BBSL from the following three perspectives. The first is to test the formalized BBSL relationships / functions to confirm that they are defined as intended. Mathematical properties are adopted for the test policy. In other words, if it is an inclusion relationship of an interval, the property of half order is proved. The second is confirmation of the descriptive ability of formalized BBSL. The BBSL study described the specifications compiled by NHTSA to confirm descriptive ability. By describing the same specifications in the formalized BBSL, make sure that it has the same description capability as the original BBSL. The third evaluates the proof ability of formalized BBSL. By proving the practical property of case block completeness, we confirm the proof ability of formalized BBSL.

From the experiments, all the properties that should be satisfied for the relations and functions of BBSL were proved. This allowed the formalization of BBSL to be defined as intended. In addition, most of the proofs were done in a small number of steps, around 10 steps. However, it took 61 steps to prove completeness, which is a practical property. The basic properties could be proved efficiently, but the practical properties were not easy to prove. It is considered necessary to review the definitions of relationships and functions. Regarding the description experiment of BBSL, it was found that the formalized BBSL has the same description ability as the original BBSL. In addition, from the experimental results, the amount of description in BBSL and the amount of description in formalized BBSL were almost clear. However, what is written in formalized BBSL is an abstract syntax, and the amount of description should be small. Therefore, it is considered that the amount of description is increasing overall. In the experiment to prove the completeness, the completeness was described as a mathematical formula and then described using the formalized BBSL. However, due to the convenience of implementation by Coq, there was a large discrepancy between the mathematical formula and the description in Coq. Therefore, the content was not such that the completeness could be described intuitively. A future issue is the implementation of a parser. This is because it is not realistic to directly describe the abstract school branch. This can be automatically generated by implementing a parser. Another problem is that it is not easy to describe the properties verified for the specifications. It is thought that this problem can be solved by defining a dedicated assertion language that describes the verified properties. In addition, by proposing a method for automatically verifying the properties described in the expression language, it is possible to verify the specifications using BBSL. It will be easier and will lead to higher quality of description specifications.