Formalizing “Adjacent Vehicles Enter Bus Lane” with Modal Logic

Huanyu Li

Department of Computer and Information Science,
Linköping University, 581 83 Linköping, Sweden
huanyu.li@liu.se

1 Introduction

1.1 Modal Logic

In propositional and predicate logic, every formula holds either true or false in any model. In another word, each formula is represented by term or terms with truth-functional operators which produce only one possible truth value for connected terms (negation $\neg$, conjunction $\wedge$, disjunction $\vee$, conditional $\rightarrow$ and biconditional $\leftrightarrow$). However, in a large number of situations, truth-functional operators cannot succeed. Considering to express something is necessarily true, something is always true in the future (namely something is true in one “world”, but maybe not true in other “worlds”). In these cases, the truth-functional operators are not enough. Thus modal operators are introduced to express all the situations in the real world. Two modal operators are defined as $\Box$ and $\Diamond$, the former can represent (necessarily, provable, required, always, definitely, etc.) and the latter can represent (possibly, consistent, permitted, sometimes, maybe, etc.) respectively [1,3]. Both $\Diamond$ and $\Box$ are not truth-functional, but intensional. That's to say, for example, $\Diamond p$'s truth-value is a function of p's intension.

1.2 Epistemic Modal Logic

Epistemic modal logic[2,3] is mainly used in reasoning about knowledge viewed as a subfield of modal logic. From a syntax perspective, the basic modal operator in epistemic modal logic is usually written in $\mathcal{K}$ and read as “it is known that”, “it is epistemically necessary that”, etc. For instance, $\mathcal{K}\alpha$ means “The agent knows α”. In this case, the assumption is the system only contains one single agent. If there are multiple agents, the representation of modal operators will be $(\mathcal{K}_1, \mathcal{K}_2, ...etc.)$ in which each $\mathcal{K}_i$ indicates a specific agent's knowledge. For instance, $\mathcal{K}_a\alpha$ means “Agent a knows α”. The dual operator of $\mathcal{K}$ holding the same relationship to $\mathcal{K}$ as $\Diamond$ to $\Box$, does not have a specific symbol in epistemic logic but represented as $\neg\mathcal{K}_a\neg\alpha$ can be read as “agent a does not know that not α”.

1.3 Kripke Semantics

Kripke Semantics [1,3] is known as relational semantics or frame semantics. Following formulas provide the basic definition of Kripke Semantics.

$\mathcal{M} = \langle W, R, V \rangle$

- $W \neq \emptyset$ is the set of all relevant states (possible worlds).
- $R \subseteq W \times W$ is the epistemic accessibility relation where wRv means "state v is epistemic accessible for the agent in the vehicle from state w".
- $V : A \rightarrow \wp(W)$ is a mapping function that assigns truth values to sentences in worlds (states).

$\mathcal{M}, w \models \alpha$ means α is true in the world w.

$\mathcal{M}, w \models \Box\alpha$ means for every possible world $w^{'}$ of w, namely $R(w, w^{'})$, $\mathcal{M}, w^{'} \models \alpha$ holds in $w^{'}$.

1.4 Standard Translation

Standard translation [1] is a method to transform formulas in modal logic into formulas in first-order logic. Basically, atomic formulas will be mapped as unary predicates and the objects in the first-order language are the possible worlds. For instance:

- $\mathcal{ST}_w(p) \equiv P(w)$, where $\mathcal{ST}_w(p)$ stands for formula p is true in world w, p is an atomic formula and $P(x)$ is true when p holds in world w.

The modal operators $(\Diamond, \Box)$ will be transformed into formulas in first-order logic based on the semantics.

- $\mathcal{ST}_w(\Box p) \equiv \forall w^{'}(R(w, w^{'}) \rightarrow \mathcal{ST}_{w'}(p))$
- $\mathcal{ST}_w(\Diamond p) \equiv \exists w^{'}(R(w, w^{'}) \wedge \mathcal{ST}_{w'}(p))$

Besides atomic formulas, other formulas in modal logic will be transformed according to firstly the atomic formulas on both sides of the logic connective follow upper rules, then the logic connectives $(\vee, \wedge, \rightarrow, \equiv)$ appearing in modal logic will be kept.

- $\mathcal{ST}_w(\neg p) \equiv \neg\mathcal{ST}_w(p)$
- $\mathcal{ST}_w(P \circ Q) \equiv \mathcal{ST}_w(P) \circ \mathcal{ST}_w(Q)$, where $\circ \in \{\vee, \wedge, \rightarrow, \equiv\}$
- $\mathcal{ST}f(p) \equiv \forall w[\mathcal{ST}_w(p)]$

2 Scenario

In many modern cities, a bus lane or bus-only lane is common in the traffic system. However, in a number of cities for instance in China, bus lanes are not utilized efficiently. The vehicles on the adjacent lanes can hardly enter bus lanes according to the traffic legislation. Otherwise, the drivers will get penalized.

Fig. 1. A traffic jam while the bus lane is "empty"

Sometimes in the traffic rush hour, the problem is the bus lane is totally empty while the adjacent lanes are in a jam as shown in Fig 1.

Now assuming the government is going to construct some rules to facilitate the use of bus lanes by allowing adjacent vehicles to enter the bus lane if the road condition at that time satisfies certain constraints.

Assuming that each vehicle has an agent that is able to verify the constraints of road condition and each bus lane has checkpoint every a certain distance (e.g. 300 meters) which means adjacent vehicles can only enter the bus lane at such points if the road condition constraints are verified. Road condition constraints contain the following examples,

- The speed of the vehicle which intends to enter bus lane should be under a specific speed.
- Forward visibility is at least 200 meters.
- There are no buses in the forward 200 meters of the checkpoint.
- There are no buses in the afterward 200 meters of the checkpoint.

Fig. 2 shows a conceptual model that a vehicle is intending enter bus lane at the checkpoint. The agent from the vehicle need verify three aspects of constraints which are current speed is under a specific value, constraints for forward road conditions and afterward conditions respectively. A final state presenting the vehicle can enter the bus lane need to satisfy that all the three aspects of constraints are verified as true. The agent in the vehicle can verify the constraints of the road condition all the time whenever before or reaching the checkpoint.

In this report, I am going to formalize the above model by means of modal logic. As the scenario is described above, the agent needs to verify constraints of road conditions. A modal operator $\mathbb{V}\alpha$ is used here. It means that "the situation specified by formula α is *verified*" is used to design the logic of the above model. The reason to choose modal logic instead of fuzzy logic to formalize this scenario is because we don't need to define a degree of when α is verified while the fuzzy approach suppose to do, namely the agent does not return the degree of how possible the road condition constraint is satisfied, it only returns true, false or uncertain.

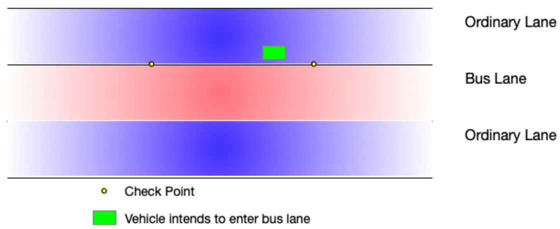

Fig. 2. Conceptual model

3 Syntax Method

3.1 The language

Let $\mathbb{V}\alpha$ informally means "α" is verified, namely "the agent in the vehicle knows that α (is true)". $\mathbb{V}\alpha$ is an intensional operator. As the assumption in this scenario, the agent may return an uncertain result. For two formulas (road condition constraints), the agent may return true and uncertain results respectively. Assuming for a certain formula, the agent may return verified formula, not verified formula or uncertain formula as results.

- $\mathbb{V}(p \to q)$ means "the agent in the vehicle knows that p implies q, namely $p \to q$ is verified".
- $\mathbb{V}p \vee \neg\mathbb{V}p$ means either the agent from the vehicle verifies p or not.
- Define $\mathbb{U}\alpha$ as $\neg\mathbb{V}\neg\alpha$, thus in natural language $\mathbb{U}\alpha$ means "$\neg\alpha$" is not verified, namely the agent from the vehicle does not know $\neg\alpha$ (is true).

α is a formula of above logic if it is of the form $\alpha := p|\neg\alpha|\alpha \wedge \beta|\mathbb{V}\alpha$

$p \in A$ where A is a set of atomic facts. For instance, p in this language could be:

- The forward visibility of the bus lane at the current checkpoint is 150 meters.
- The speed at current checkpoint is 35 km per hour.
- There is no bus coming in the 200 meters afterward in the bus lane at current checkpoint.

Valid properties of this language contain for instances:

- **D**: $\mathbb{V}a \to \mathbb{U}a$. According to the description of the scenario, this property means for a verified road condition constraint (a), there exist at least one possible state after current state can still hold this road condition constraint (a).
- **T**: $\mathbb{V}a \to a$.
- $\mathbb{V}a \to \mathbb{V}\mathbb{U}a$.

Invalid properties contain:

- **4**: $\mathbb{V}b \to \mathbb{V}\mathbb{V}b$. The semantic for this property is for every state that b is verified, it can imply that b is also verified in the following possible states. However this is not true in the scenario, for instance, after the vehicle gets forward road condition constraints are verified before the checkpoint, the road condition may change when the vehicle reaches the checkpoint. Thus at that point, forward road condition constraints will be not verified which is controversial to this property.
- $\mathbb{U}b \to \mathbb{V}\mathbb{U}b$.

3.2 Kripke Model

To formalize the model with Kripke semantic, assuming three atomic formulas are:

- p stands for the speed of the vehicle at the checkpoint is under a specific value.
- q stands for the constraints relevant to the forward 200 meters' road condition are satisfied.
- r stands for the constraints relevant to the afterward 200 meters' road condition are satisfied.

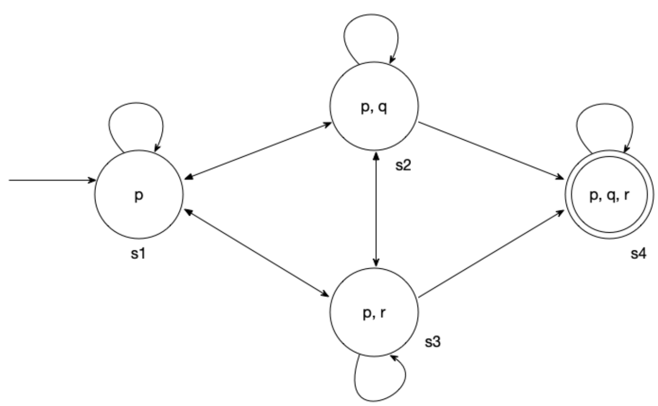

Fig. 3. Possible states

Suppose an initial state (s1) that the driver of a vehicle in the adjacent lane intends to enter the bus lane and the agent in the vehicle has verified p but does not verify q or r yet. Further, there are two possible states after s1, the first one (s2) is the agent first gets q is verified thus it knows both p and q but does not know r yet, the second possible state (s3) describes the agent first gets r is verified thus it knows both p and r but does not know q yet. From the view of the agent, these two states are equivalent. State pairs (s1, s2) and (s1,s3) are also equivalent pairs respectively assuming when the agent verified the forward or after road condition constraints are true before the checkpoint may be false when the vehicle reaches the checkpoint. For the former state (s2), if the agent verifies r, then it goes to a final state (s4) where p, q, r hold meaning the vehicle can enter the bus lane. Similarly, if the agent verifies q in the above latter state (s3). Overall, each state is accessible from itself as the agent holds the same knowledge before the checkpoint and reaching the checkpoint.

For the valid property presented in subsection 3.1 according to the above semantics, it is obvious that atomic formula p satisfies every property.

- **D**: $\mathbb{V}a \rightarrow \mathbb{U}a$. This is obviously successful for every formula as for each world that the specific formula is true, there exists a possible world also holding the formula as true.
- **T**: $\mathbb{V}a \rightarrow a$. This is obviously successful for every formula as for each world that the specific formula is true, the formula is true within the specific world.
- $\mathbb{V}a \rightarrow \mathbb{V}\mathbb{U}a$. Taking formula q as an instance, the left side based on the property will be (s2,s4). Taking s2 as $\mathbb{U}a$, It is obvious that every accessible state from s2 satisfies formula q.

3.3 Translation into first-order logic

Taking one valid property ($\mathbb{V}a \rightarrow \mathbb{U}a$) in subsection 3.1 as an example, the transformation steps are shown as follow:

$$
\begin{aligned}
&\mathcal{ST}f(\mathbb{V}a \rightarrow \mathbb{U}a) = \\
&\qquad \forall w[\mathcal{ST}_w(\mathbb{V}a \rightarrow \mathbb{U}a)] = \\
&\qquad \forall w[\mathcal{ST}_w(\mathbb{V}a) \rightarrow \mathcal{ST}_w(\mathbb{U}a)] = \\
&\qquad \forall w[[\forall w^{'}(R(w, w^{'}) \rightarrow \mathcal{ST}_{w'}(a))] \rightarrow [\exists w^{''}(R(w, w^{''}) \wedge \mathcal{ST}_{w''}(a))]] = \\
&\qquad \forall w[[\forall w^{'}(R(w, w^{'}) \rightarrow A(w^{'}))] \rightarrow [\exists w^{''}(R(w, w^{''}) \wedge A(w^{''}))]] = \\
&\qquad \forall x[[\forall y(R(x, y) \rightarrow A(y))] \rightarrow [\exists z(R(x, z) \wedge A(z))]] = \\
&\qquad \forall x \forall y \exists z[[R(x, y) \rightarrow A(y)] \rightarrow [R(x, z) \wedge A(z)]] = \\
&\qquad \forall x \forall y \exists z[[\neg R(x, y) \vee A(y)] \rightarrow [R(x, z) \wedge A(z)]] = \\
&\qquad \forall x \forall y \exists z[\neg[\neg R(x, y) \vee A(y)] \vee [R(x, z) \wedge A(z)]] = \\
&\qquad \forall x[\forall y[R(x, y) \wedge \neg A(y)] \vee \exists z[R(x, z) \wedge A(z)]] = \\
&\qquad \forall x[\forall y \neg A(y) \vee \exists z R(x, z)] = \\
&\qquad \forall x \exists z[R(x, z)]
\end{aligned}
$$

Reflecting in figure 3, for each state in the model, there exists at least one accessible world. A translation for invalid property $\mathbb{V}b \rightarrow \mathbb{V}\mathbb{V}b$ is shown as follow:

$$
\begin{aligned}
&\mathcal{ST}f(\mathbb{V}b \rightarrow \mathbb{V}\mathbb{V}b) = \\
&\quad \forall w[\mathcal{ST}_w(\mathbb{V}b \rightarrow \mathbb{V}\mathbb{V}b)] = \\
&\quad \forall w[\mathcal{ST}_w(\mathbb{V}b) \rightarrow \mathcal{ST}_w(\mathbb{V}\mathbb{V}b)] = \\
&\quad \forall w[[\forall w^{'}(R(w,w^{'}) \rightarrow \mathcal{ST}_{w'}(b))] \rightarrow \forall w^{''}[R(w,w^{''}) \rightarrow \mathcal{ST}_{w''}(\mathbb{V}b)]] = \\
&\quad \forall w[[\forall w^{'}(R(w,w^{'}) \rightarrow B(w^{'}))] \rightarrow \forall w^{''}[R(w,w^{''}) \rightarrow \forall w^{'''}[R(w^{''},w^{'''}) \rightarrow \mathcal{ST}_{w'''}(b)]]] = \\
&\quad \forall w[[\forall w^{'}(R(w,w^{'}) \rightarrow B(w^{'}))] \rightarrow \forall w^{''}[R(w,w^{''}) \rightarrow \forall w^{'''}[R(w^{''},w^{'''}) \rightarrow B(w^{'''})]]] = \\
&\quad \forall w[[\forall x(R(w,x) \rightarrow B(x))] \rightarrow \forall y[R(w,y) \rightarrow \forall z[R(y,z) \rightarrow B(z)]]] = \\
&\quad \forall w \forall x \forall y \forall z[[(R(w,x) \rightarrow B(x))] \rightarrow [R(w,y) \rightarrow [R(y,z) \rightarrow B(z)]]] = \\
&\quad \forall w \forall x \forall y \forall z[[(\neg R(w,x) \vee B(x))] \rightarrow [R(w,y) \rightarrow [R(y,z) \rightarrow B(z)]]] = \\
&\quad \forall w \forall x \forall y \forall z[[\neg(\neg R(w,x) \vee B(x))] \vee [R(w,y) \rightarrow [R(y,z) \rightarrow B(z)]]] = \\
&\quad \forall w \forall x \forall y \forall z[[(R(w,x) \wedge \neg B(x))] \vee [R(w,y) \rightarrow [R(y,z) \rightarrow B(z)]]] = \\
&\quad \forall w \forall y \forall z[R(w,y) \rightarrow [R(y,z) \rightarrow B(z)]]] = \\
&\quad \forall w \forall y \forall z[\neg R(w,y) \vee [\neg R(y,z) \vee B(z)]]] = \\
&\quad \forall w \forall y \forall z[\neg(R(w,y) \wedge R(y,z)) \vee B(z)]]] = \\
&\quad \forall w \forall y \forall z[(R(w,y) \wedge R(y,z)) \rightarrow B(z)]]]
\end{aligned}
$$

Considering the definition of R in Kripke Semantics, during the translation, we have known that b is true in every possible states and we have $\forall w \forall y \forall z[(R(w,y) \wedge R(y,z)) \rightarrow B(z)]]]$ in above translation. As a result, we can derive that world z is accessible from world w. Thus, the final translation is as follow:

$$\mathcal{ST}f(\mathbb{V}b \rightarrow \mathbb{V}\mathbb{V}b) = \forall w \forall y \forall z[(R(w,y) \wedge R(y,z)) \rightarrow R(w,z)]]]$$

After applying the standard translation on $\mathbb{V}b \rightarrow \mathbb{V}\mathbb{V}b$, we know that this formula reflects the transitivity in the model. However, as figure 3 shows, the scenario and model does not satisfy the property of transitivity. Taking states s1, s2 and s4 as examples, there exists $R(s1, s2)$ and $R(s2, s4)$, but $R(s1, s4)$ does not hold which is against transitivity.

3.4 Database querying

Taking the standard translation result $\mathcal{ST}f(\mathbb{V}a \rightarrow \mathbb{U}a) = \forall x \exists z[R(x,z)]$ as an instance. Supposing the final solution of this model contains a table that has the records of the agent from two continuous states as shown in table 1. This property satisfies the reality in the table that, for each possible latter state, the agent has returned a valid state as in the table there is no NULL value. So the query meets this property will be for a given value of former state's result, return the non-NULL value from the latter state's result.

Table 1. An example of a table in the database

former state's result	latter state's result
100	100
100	101
100	110
110	100
110	101
110	110
110	111
101	100
101	101
101	110
101	111
111	111

*For the data in each cell, there are three digits which represent the values of atomic formulas processing by the agent ($\mathbb{V}$). For example, 100 means atomic formula p is verified but q and r are not verified..

4 Reflection

This report describes the process of designing a logic with operator $\mathbb{V}$ with the aim at formalizing "adjacent vehicles enter a bus lane" behavior with modal logic. Table 2 shows the places in the report where questions are addressed.

Table 2. Questions Reflection

Question	Places in the report
1	Section 2
2	Subsection 3.1
3	Subsection 3.1
4	Section 2
5	Subsection 3.1
6	Subsection 3.2
7	Subsection 3.2
8	Subsection 3.3
9	Subsection 3.3
10	Subsection 3.4

References

1. Blackburn, P., Van Benthem, J.: 1 modal logic: a semantic perspective. In: Studies in Logic and Practical Reasoning, vol. 3, pp. 1–84. Elsevier (2007)
2. Hendricks, V., Symons, J.: Epistemic logic. In: Zalta, E.N. (ed.) The Stanford Encyclopedia of Philosophy. Metaphysics Research Lab, Stanford University, fall 2015 edn. (2015)
3. Szałas, A.: Cugs: Logic ii lecture slides (2018, 12)