Probabilistic Conditional System Invariant Generation with Bayesian Inference: Appendix

[Anonymized]
Email
Affiliation

In this Appendix with provide additional study data and explanations that we could not fit in the paper submission.

I. Autonomous Vehicle Invariants

This analysis is relevant to the second study in the paper.

Table I includes the top 10 invariants generated for the autonomous vehicle. We briefly comment on them here.

Invariant *P(Brake>0 — Mode=autonomous Throttle==0 Event=pedestrian detected TrustChange>0)* on row 1 had the highest surprise ratio, possibly due to the specific behavior the autonomous controller exhibited in the presence of a pedestrian and the trust-building effect it had on the human in the loop.

Invariant *P(Throttle==0 | Mode = autonomous Event=pedestrian detected)* on row 2 had a similarly high surprise ratio, with posterior likelihood of 1.0 indicating that when the car is autonomously controlled and a pedestrian is in the roadway, the throttle is no longer engaged.

This has a slightly higher probability than *P(Brake>0 | Mode=autonomous Throttle==0 Event=pedestrian detected TrustChange >0)*, likely because the throttle must be disengaged before brake can be engaged, and the brake may be engaged for multiple reasons, such as a different event or a curve in the road. The inclusion of predicate $TrustChange>0$ in the givens was surprising as well, as it indicates that an increase in trust in conjunction with detecting a pedestrian is correlated with a subsequent application of the brakes. Note that this is not a causal relationship, as the autonomous driving algorithm does not react to changes in trust from the human in the loop.

Dealing with incidents on the road is essential to demonstrate realistic safety behaviors in the autonomous driving scenario. Invariants given *Event=Pedestrian detected* characterize the performance of the simulated car when handling the incident of pedestrian crossing the road. The car decreases the velocity by applying brake and easing the throttle, but no wheel change is closely associated according to the model selection algorithm. It fits the expectation that the car tends to slow down, rather than changing lanes to bypass the pedestrian. In larger models, both $WheelChange>=20$ and $WheelChange<20$ predicates present with no significant change to the posterior likelihood, showing that the posterior likelihood is more strongly conditioned on a change in $Throttle$.

On the other hand, invariants given *Event=Cyclist detected* shows that the car performs a steep turn and unexpectedly accelerates in order to avoid the cyclist. Outside of the top ten, similar invariants can be found involving $Event=Truck$, which shows that the car again unexpectedly accelerates when passing the incoming truck on the other lane. These invariants present trends that the designers were unaware of and provide direct guidance on improving the system design when handling incidents.

In the autonomous driving scenario, alarms are important to alert drivers and get their attention to possible incidents. We intentionally injected false alarms to test the car's performance. The expected invariant $P(-20<=WheelChange<=20|Event=FalseAlarm)$ tells that the car maintains slight change of the wheel, while the unexpected invariant $P(VelocityChange<0|Event=FalseAlarm)$ indicates a velocity decrease. Recall that false alarm is an auditory alarm when there is no real incident on the road. Even though the false alarm does not cause the car to react too much on the wheel, it causes the car to slow down to check if there is any incident. False alarms improve the safety of the system by detecting potential hazards more conservatively, though too many false alarms may lead to driver's alarm fatigue.

Invariants related to $Mode$ provide the information during manual driving or autonomous driving. Invariant *P(TrustChange>0 | Brake==0 Mode=manual Throttle>0 Event=None)* on row 5 shows that human drivers tend to increase their trust following "normal" operation in manual mode. This also indicates that they subsequently leave manual mode, as it is not possible to raise trust in manual mode.

$P(Mode=Manual|Throttle=0)$ with a ratio of 3.986 and $P(Mode=Autonomous|Throttle=0)$ with a ratio of 0.481 tells that given throttle is not used, the car is more likely to be in manual driving mode. These invariants shows that human drivers tends to be unsatisfied by the speed of the autonomous driving and they are confident of driving manually. This can help the designer of the system to understand human driver's behavior and adjust the performance of the system.

Trust affects human drivers' reliance on the sys-

TABLE I: Driving Invariants

Invariant	Posterior	Original prior	Surprise Ratio	Explanation
P(Brake>0 | Mode=autonomous Throttle==0 Event=pedestrian detected TrustChange>0)	0.97	0.04	21.85	When mode is autonomous, throttle is not engaged, a pedestrian is in the roadway, and trust has increased, brake is likely engaged.
P(Throttle==0 | Mode=autonomous Event=pedestrian detected)	1	0.1	10.26	When mode is autonomous and a pedestrian is in the roadway, throttle is likely not engaged.
P(Mode=manual | Throttle==0 Event=None)	0.95	0.15	6.40	When throttle is not engaged and nothing is detected in the roadway, mode is likely manual.
P(WheelChange<20 | Mode=autonomous Event=None)	0.53	0.50	2.91	When mode is autonomous and nothing is detected in the roadway, wheel angle is likely changing slowly.
P(TrustChange>0 | Brake==0 Mode=manual Throttle>0 Event=None)	0.31	0.12	2.695	When brake is not engaged, mode is manual, throttle is engaged, and nothing is detected in the roadway, trust is likely increasing.
P(WheelChange>=20 | Brake==0 Mode=autonomous Throttle>0 Event=cyclist detected TrustChange¡0)	0.90	0.5	1.79	When brake is not engaged, mode is autonomous, throttle is engaged, a cyclist is in the roadway, and trust increased, wheel angle is likely changing quickly.
P(TrustChange<0 | Brake==0 Mode=autonomous Throttle>0 Event=None WheelChange>=20)	0.15	0.12	1.21	When brake is not engaged, mode is autonomous, throttle is engaged, nothing is detected in the roadway, and wheel angle is changing quickly, trust is likely to decrease.
P(Throttle>0 | Mode=autonomous Event=None)	1	0.90	1.11	When mode is autonomous and nothing is detected in the roadway, throttle is likely engaged.
P(Brake==0 | Mode=autonomous Event=None)	1	0.96	1.05	When mode is autonomous and nothing is detected in the roadway, brake is likely not engaged.
P(TrustChange==0 | Brake==0 Mode=autonomous Throttle¿0 Event=None WheelChange>=20)	0.77	0.77	1.01	When brake is not engaged, mode is autonomous, throttle is engaged, nothing has been detected in the roadway, and wheel angle is changing quickly, trust is likely to remain constant.

tem. *P(TrustChange < 0 | Brake==0 Mode=autonomous Throttle>0 Event=None WheelChange >= 20)* on row 7 tells that when throttle is engaged and the wheel angle is changing quickly, the trust level is more likely to decrease. The absence of an $Event$ seems to indicate this is a perceived safety issue on the part of the human in the loop.

However, *P(TrustChange==0 | Brake==0 Mode=autonomous Throttle>0 Event=None WheelChange >= 20)* on row 10 shows us that the human in the loop is more likely to leave trust unchanged under those same conditions. This seems to indicate that there are latent variables not being accounted for, possibly in the system or on the part of the user. Fully understanding the trust evolution contributes to a trustworthy system.

II. Preliminary Results on Inference Computation Cost

We explore *What is the cost of generating conditional invariants?*. We assess cost in terms of the time to generate the invariants as a function of the number of predicates and trace length.

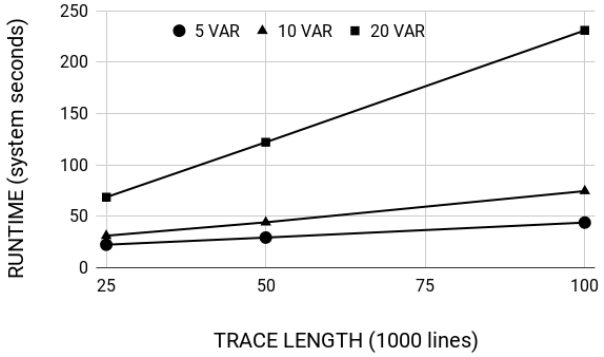

Fig. 1: Runtime vs. trace length and number of predicates.

We briefly assess the dimensions of the space of invariants to explore and the associated inference cost. Figure 1 plots the runtime cost in system seconds when executing the inference engine on traces of three different lengths (25K, 50K, 100K) produced by the Drone ISR when three different sets of predicates are explored (5, 10 and 20 variables of $Range$ type

with two predicates each). Runtime tests were performed on a containerized Linux box with an x86_64 AMD FX-8120E 3.1GHz 8-core processor.

As the graph shows, the runtime of the engine depends on both trace length and predicate complexity, but the influence of the number of variables seems to dominate the space to explore as it grows exponentially when the variables are considered as both outcomes and givens. As the number of variables grows, a developer can control this cost through the specification of the predicate space provided to the engine by specifying whether a variable is to be explored as a given or as an outcome, or more restrictively by aiming for particular pairs of variable predicates.