The Chicken Can Cross The Road

Bohan Li and Dongho Choi

December 9, 2014

1 Abstract

This paper is motivated by the age-old question: why did the chicken cross the road? In this paper we discuss the challenges of developing and modelling a cyber-physical system to help a chicken (a robot or rover) safely pass through a field (road) of moving obstacles (cars) to reach a predetermined goal (the other side of the road). In this project, we construct hybrid *dL* programs that capture simplified versions of this problem. We then prove the corresponding safety condition that the chicken will not collide with any cars regardless of whatever choices it may make, and we also prove that in certain, realistic situations, the chicken will always have a way of crossing the road without getting hit by any cars. Parts of the proofs are automated by a theorem prover called Keymaera, while the rest is proved with traditional manual sequent proofs.

2 Introduction

The inspiration behind this project came from the classic joke: Why did the chicken cross the road? We took it a step further and began imagining what it would look like if the chicken actually decided to cross the road. Thinking about the question in terms of a cyber-physical problem, the chicken could be represented as a robot or rover that could control it's speed and direction. The road could be represented using as a set of lanes with cars moving in one direction within each lane. As we explored this idea, we realized that the system we were developing belonged to category of cyber-physical systems classified by a single object moving through a field of predictable (moving) obstacles. Real-world problems such as flying a drone in airspace crowded with commercial planes and steering a spaceship through an asteroid belt fall under this category of cyber-physical problems. If we could solve the chicken crossing the road problem, we could gain much insight into solving the multitude of problems in this genre. After considering this viewpoint, the question of whether or not the chicken could cross the road quickly developed into a very interesting and challenging problem.

A hybrid program consists of a set of preconditions, control decisions made by a controller (in this case, the chicken), a physics model, and some property to prove (usually, a safety property of the controller). The program assumes the preconditions are met and starts of with the controller. The controller, once it has the chance, makes control decisions to assign values to key variables and passes control to the physical model. The physics model part utilizes differential equations to simulate the effect of real-world physics on the controller and obstacles. The interaction between physics and logic usually can take one of two forms: event-triggered or time triggered. Event-triggered programs give control to the controller based on some condition defined in terms of existing program variables. This condition usually marks some significant event happening in the physical world. Time-triggered programs involve a timer ticking at a constant rate, and they give control to the controller once the timer hits a certain threshold. Both of these types of programs have evolution domain constraints which enforce basic properties of the model. Finally, the safety property at the end must after any possible run of the program; this, in a sense, shows that during all times of the program, the controller will be in a safe position. We use sequent proof rules and a theorem prover called Keymaera to prove that the safety property does indeed after any possible run of the program.

2.1 Related Work

As mentioned above, there are quite a few different problems that can be classified in the same category as ours. For example, one general question for which we can gain insight on is: can an object pass through a field of moving obstacles if it knows the paths they will take? There are, of course, key differences between such general questions and our simplified question, which mostly involve the motion of the object and obstacles.

One application in which we can see differences from our problem is the problem of safely moving a drone through a crowded airspace.

The most apparent difference lies in the fact that the drone exists in a 3-dimensional space while our chicken problem generally lies in 2 dimensions. This difference will obviously affect the differential equations governing the motion of the drone and obstacles as well as the complexity of collision calculation.

Another difference is the assumption one makes in the drone problem. The drone will most likely be moving at a very high speed, so it's acceleration will be a relatively small percentage of it's current speed, and thus would be less maneuverable with regard to speed and raw acceleration. However, in our simplification, we allow the chicken to make sharp, instantaneous changes in velocity (including direction). This assumption is not unrealistic, since chickens do travel at relatively low speeds, where sharp turns and speed changes are feasible. This means that the drone need more foresight and will consider events that are much further in the future, while the chicken would be able to get away with making only short-term decisions (assuming, of course, the chicken can react fast enough).

Yet another major difference to the complexity of the obstacle paths and the existence of "safe zones", where a safe zone is any area of space that is not occupied by an obstacle. Safe zones can be used to "reset" the problem in a sense. Safe zones are important for solving the global problem by essentially inductively reasoning about the problem: we can focus on traveling from one safe zone to the next, turning a very large problem into many smaller ones. Although the paths of the obstacles may be predetermined and the object can observe the spaces occupied by these paths and determine the location of any safe zones in either problem, the complexity of the paths allowed by the object (drone or chicken) or obstacles (planes or cars) vary quite greatly. Further, the drone problem will have a significant portion of the airspace covered in safe zones, while on the other hand, the chicken problem can have very little. The shape, periodicity and the spacing of the safe zones can also affect the complexity of calculation required to determine safety of the object.

Another simpler problem that is helpful to consider is Lab 4*b* from the 15-424 curriculum. This involves a robot traveling along a path while avoiding a moving obstacle. The robot knew the obstacle's path and could determine the obstacle's location at all times. The strategy the robot should adopt is to accelerate if it is safe to do so or to brake if it detects that a collision will occur if it keeps moving; however, the moving obstacle had great freedom in its choices of movement, which made it very difficult to develop a strategy such that the robot will always have a safe choice of movement. Our problem will involve a similar type of control decision; but, by simplifying the assumptions of the obstacles and the chicken (the chicken is allowed to instantaneously change velocity and is only allowed to move at a constant velocity or hover instantaneously in the air in place, and the cars move only at a constant velocity), our problem becomes more tractable.

3 The Model

In this section, we take a more in-depth look into the specifics of solving our problem.

3.1 Cars

We modelled each car to be a 3-dimensional box having with w, length l and height h. See the figure for a graphical illustration. The position of the car is taken to be the front, bottom, center of the car, and is denoted by (c_x, c_y, c_z), where generally $c_z = 0$ since cars stay on the ground. We believe this to be a reasonable simplication of a car, since in the real world, the base of a car closely resembles a rectangle.

The model of our car initially started off as simple as possible. We assumed the car to have zero length or height, i.e., $l = 0$ and $h = 0$. This allowed us to get the foundations of our program proven before

we moved onto more complicated arithmetic. Afterwards, we considered cases where cars were allowed to have non-zero length and height. During the process of this generalization, we faced some challenges, which is discussed in the section on the Distributed Model.

Each car has initial positive velocity u. The velocity is strictly in the positive x direction, which models cars travelling on a one-direction road, so that the differential equation modelling the position of a car is simply $c'_x = u$. In the general case, we do not specify any restriction on the car's velocity; however, later, we will discuss restriction on car velocity to model speed limits, and other similar practical restrictions to allow the chicken to cross the road. This velocity needs to be bounded so that the cars do not travel fast enough to eliminate any possibility for the chicken to cross. This computation for these restrictions will be discussed in the corresponding sections.

We assumed the cars travel at a constant velocity. We also believe that this assumption is not unreasonable, since on long stretches of road, besides exceptional cases, cars tend to travel at a similar, constant speed.

Each car is allowed to begin at any position as long as it begins strictly inside the road. Specifically, we enforced that $0 < c_y < H$. In practical cases, which will be discussed in later sections, further restrictions may be placed on the cars that allow for gaps between cars. Such additions model the real world, since cars will travel at a safe distance apart from other cars. Further, such gaps give the chicken opportunities to make progress towards the other side of the road, which adds to the feasibility of the problem.

Further restrictions on the chicken are placed depending on which overall model of the problem we were considering, which will be discussed below.

3.2 The Chicken

The chicken will be represented as a point (k_x, k_y, k_z). This is justified by the fact that a chicken is quite small compared to a car; also, more accurate interpretations of the problem can easily be accommodated by simply make the cars bigger to account for the size of the chicken. Furthermore, it allows us to focus on the fundamental proof structure without having to worry about messy calculations.

The chicken's initial position will be $(0, 0, 0)$, which is simply an arbitrary point along the length of the road. This starting point makes some of the proofs of the hybrid programs easier, and does not detract from the generality of the problem.

The chicken's movement will vary based on the constraints we set. The complexity of our models and proofs is largely based on the mobility of the chicken. The most basic chicken movement is the ability to accelerate forwards. This is the only option we allow in the event-based hybrid program where the chicken waits for the right moment to sprint across. We eventually made a simplification by ignoring the chicken's top speed, but the main idea was captured nonetheless.

In the remaining time-triggered models, the chicken is allowed to move in any direction in the Cartesian plane. Instead of accelerating to change velocities, we allow the chicken to make instantaneous changes in velocity v. This assumption is not unrealistic, since chickens do travel at relatively low speeds, where sharp turns and speed changes are feasible, i.e., a chicken can accelerate from a relatively small velocity to another relatively small velocity rather quickly. In practical considerations later, we bound the velocities that the chicken can move at to be low enough that these instantaneous changes are more in line with real-world scenarios.

Additionally, we give the chicken the option to instantly jump into the air above the cars and hover in place, setting all directional velocities to 0. This is probably the least realistic assumption we make; however it somewhat justified by the fact that other birds (not necessarily chickens) can place themselves in the air quite quickly. Finally, this option does not make the problem of crossing the road any hard, because while the chicken can hover in safety, it cannot make any progress towards the other side of the road, i.e., safety does not imply efficiency, which is just as important in this problem.

Finally, the time limit we use in our time-based controller can be viewed as the reaction time of the chicken.

Further restrictions on the cars are placed depending on which overall model of the problem we were considering, which will be discussed below.

3.3 The Problem Model

One of the first steps of the project was to determine how to transition the real world problem into a logical one that we can prove. We required a way to somehow represent an infinite stream of cars on a road.

One simple way of representing this is to have a single car repeatedly pass through the same stretch of space. To accomplish this, we first considered having circular tracks for each car. The rings would stack around each other, so that the rings are concentric, and the chicken would begin in the center and move outward. However, this approach leads to uneven lengths for the tracks, which made it difficult to capture a sense of a regular flow of traffic through which the chicken must maneuver.

One way we attempted to fix this was to add another dimension to the problem.Instead of stacking the tracks around each other, we stacked them on top of each other. The chicken would then move upwards through the stack, as an elevator would ascend through floors. The tracks are now identical, and each car is representing a uniform stream of cars. This model is what we started out with; however we quickly realized that the non-physical third dimension as well the complexity of circular motion made it difficult to model our problem with successful proofs.

3.3.1 Teleportation Model

Eventually, we decided to keep and maintain a linear and overall 2-dimensional approach. One idea we considered was to model the road as a finite rectangle in the xy-plane of length L and width H on which cars travel in the positive x direction. We then had cars "teleport back to the beginning (the left) of the road once it reached the end. This would model a consistent and fluid flow of traffic on a road with only using one car per lane. With this, we were able to use the Cartesian coordinate system to model the problem. See Figure 2 for a graphical illustration of this model.

Cars in this model were modelled to have zero length. We did not specifically consider the scenario where in real life the cars would travel in disjoint lanes in the road because this was only a specific instance of the more general situation which we considered instead. The cars can start anywhere within the boundaries of the road. Furthermore, in this version, it is okay for cars to overlap; we took this general approach because restricting the position of the cars only makes it safer for the chicken to cross the road. In other words, proving the safety property under a more general model will end up proving safety for any specific instances of the model; hence, in future models, we decided to adopt this mentality, and then consider efficiency conditions for specific instances of the models.

We modelled the chicken as described before. The chicken will move the the positive y-direction to cross the road. The chicken is also allowed a non-zero x-component in its velocity. Of course, if the chicken travels past the left end of the road, then it will instantly teleport to the right end of the road; similarly, if the chicken travels past the right end of the road, then it will instantly teleport to the left end of the road.

In this version, we left out the scenario where in real life the cars would travel in disjoint lanes in the road because this was only a specific instance of the more general situation we considered. The cars can start anywhere they want within this boundary, meaning that our original lane separated case is still possible. However, if we allow cars to overlap each other, we have made a variety of different scenarios possible. Proving the safety property under this model will end up proving safety for all the configurations that this model allows. We decided to adopt this model initially and considered about how the original problem fit this model.

3.3.2 Distributed Model

After analyzing the Teleportation Model, we attempted to generalize the definition of the car, by giving it non-zero length. However, we quickly realized that the arithmetic required to determine whether the path of the chicken would cross the path of a continually teleporting car of nonzero length was extraordinary difficult with many edge cases. We also desired to relax the assumption that cars travel on a road according to a constant flow of traffic. Finally, generalizing the Teleportation Model to accommodate more than one car was quite forbidding and impractical.

However, we noted that the logic required in path intersection between the chicken and each car and the

physics evolution of each car was essentially the same for each car. Using this insight, in the end, we decided to take a distributed approach, where we could apply quantified differential logic QdL to model every (potentially infinitely many) cars by creating a sort (or type) C for cars and applying logic generally to all objects of sort C. This sort of distributed approach works because the physics evolution and any logic involving one car could applied in exactly the same way for any other car.

Cars in this model are allowed to have non-zero length and height, which is a big improvement over the previous model. As before, we did not specifically consider the scenario where in real life the cars would travel in disjoint lanes in the road and we allowed cars to overlap because we considered the general scenario instead. The cars can start anywhere within the boundaries of the road.

We modelled the chicken as described before. However, when making a jump and hover move, we ensure that the chicken jumps to a height greater than the height h of the cars.

In this version, we left out the scenario where in real life the cars would travel in disjoint lanes in the road because this was only a specific instance of the more general situation we considered. The cars can start anywhere they want within this boundary, meaning that our original lane separated case is still possible. However, if we allow cars to overlap each other, we have made a variety of different scenarios possible. Proving the safety property under this model will end up proving safety for all the configurations that this model allows.

4 Hybrid Programs and Safety Proof Details

We present the major hybrid programs used to prove the safety of the different models of our problem.

4.1 Teleportation Model

We introduce some abbreviations:

$$Q_1 \equiv \mathbf{k} = 0 \land \mathbf{v} = 0$$
$$Q_2 \equiv 0 < c_y - \frac{w}{2} \land c_y + \frac{w}{2} < H \land c_z = 0 \land u_x > 0$$
$$A \equiv Q_1 \land Q_2$$
$$B \equiv H > 0 \land w > 0 \land h \geq 0 \land l = 0 \land T > 0$$
$$\alpha \equiv k_z := h + 1; v_x := 0; v_y := 0; v_z := 0$$
$$\beta_1 \equiv k_z := 0; v_x := *; v_y := *; v_z := 0; ?(v_y > 0)$$
$$Q_3 \equiv k_x < c_x - l \lor c_x < k_x \lor t_1 > T$$
$$\beta_2 \equiv \gamma_1; \gamma_2; \mathsf{if}\ v_x = u_x\ \mathsf{then}\ ?Q_3\ \mathsf{else}\ \gamma_3; \gamma_4; \delta_1; \delta_2\ \mathsf{fi}$$
$$\beta \equiv \beta_1; \beta_2$$
$$\gamma_1 \equiv t_1 := \frac{c_y - \frac{w}{2} - k_y}{v_y}$$
$$\gamma_2 \equiv t_2 := \frac{c_y + \frac{w}{2} - k_y}{v_y}$$
$$\gamma_3 \equiv s_1 := \frac{c_x - k_x}{v_x - u_x}$$
$$\gamma_4 \equiv s_2 := \frac{c_x - l - k_x}{v_x - u_x}$$
$$\delta_1 \equiv \mathsf{if}\ s_1 > s_2\ \mathsf{then}\ q := s_1; s_1 := s_2; s_2 := q\ \mathsf{fi}$$
$$\delta_2 \equiv ?(t_2 < s_1 \lor s_2 < t_1 \lor t_1 > T \lor s_1 > T)$$
$$\epsilon \equiv \{c_x' = u_x, \mathbf{k}' = \mathbf{v}, t' = 1, t \leq T\}$$
$$\chi \equiv t := 0; (\alpha \cup \beta); \epsilon$$

Then the hybrid program for the Teleportation Model is $\mu_T \equiv \chi^*$. The safety condition for this program is

$$S = \neg(0 \leq c_x - k_x \leq l \land |c_y - k_y| \leq \frac{w}{2} \land 0 \leq k_z - c_z \leq h),$$

and the initial conditions are $A \land B$.

Firstly, let us analyze this program. The chicken first gets a choice. The chicken can choose α, in which case, the chicken instantaneously jumps to a height of $h + 1$ to avoid all cars, but does not get to make progress by setting its velocity to 0.

On the other hand, the chicken can choose β. Choice β works as follows. The chicken first settles back onto the ground. Then it chooses velocities v_x and v_y (its up component v_z will be 0) such that in the worst case scenario, if the chicken travels for time T, it will not collide with the car.

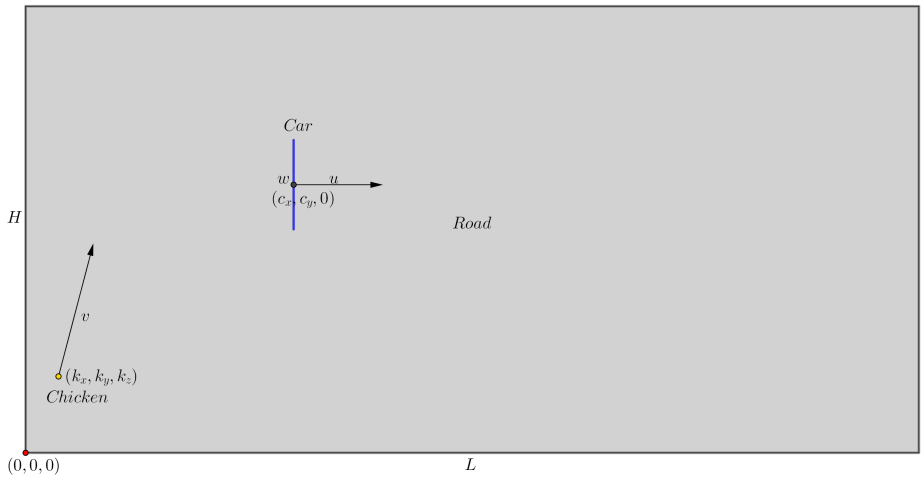

Figure 2: Teleportation Model

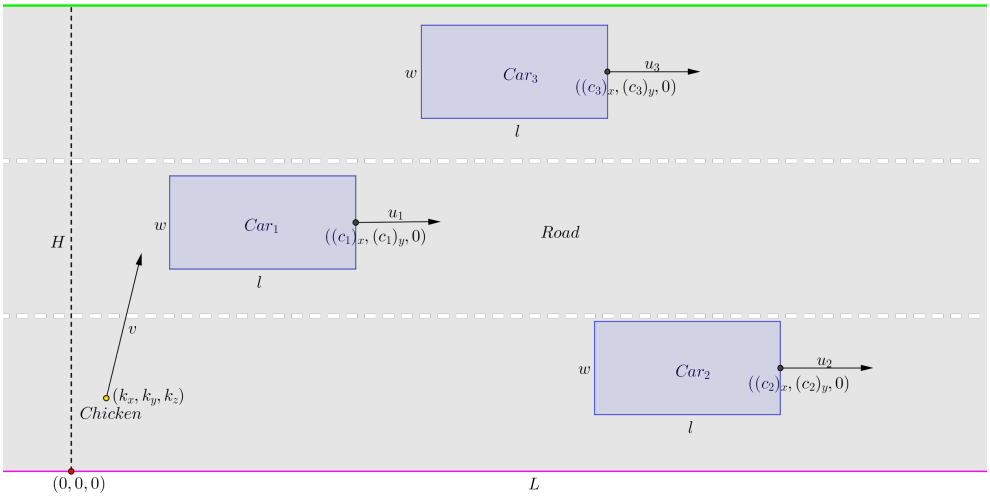

Figure 3: Distributed Model

Suppose $v_x = u_x$. Then that condition is given by formula Q_3, or

$$k_x < c_x - l \lor c_x < k_x \lor t_1 > T,$$

where

$$t_1 = \frac{c_y - \frac{w}{2} - k_y}{v_y}$$

from the assignment in γ_1 in β_2. Note that t_1 is equal to the time it would take for the chicken to reach the path in which the car is travelling if the chicken were to continue travelling at velocity (v_x, v_y, v_z) forever, i.e., the time it takes for the chicken to reach the interval $\left[c_y - \frac{w}{2}, c_y + \frac{w}{2}\right]$. Note that t_1 is uniquely and well defined (t_1 could be negative) because the chicken is travelling at a constant velocity. If $t_1 > T$, then there is no chance that the chicken will collide with the car, since it will not even cross the path the car is taking. Further, if $k_x < c_x - l$ or $c_x < k_x$, then the chicken can never collide with the car since its x-velocity is the same as that of the car.

On the other hand, suppose $v_x \neq u_x$. Then we first compute

$$t_1 = \frac{c_y - \frac{w}{2} - k_y}{v_y}$$
$$t_2 = \frac{c_y + \frac{w}{2} - k_y}{v_y}$$
$$s_1 = \frac{c_x - k_x}{v_x - u_x}$$
$$s_2 = \frac{c_x - l - k_x}{v_x - u_x}.$$

Note as before t_1 is equal to the time it takes for the chicken to reach the path in which the car is travelling if the chicken were to continue travelling at velocity (v_x, v_y, v_z) forever; similarly, t_2 is the time it takes for the chicken to enter and then exit the path in which the car is travelling if the chicken were to continue travelling at velocity (v_x, v_y, v_z) forever, i.e., then time it takes for the chicken to enter and then exit the interval $\left[c_y - \frac{w}{2}, c_y + \frac{w}{2}\right]$. Note that t_1 and t_2 are uniquely and well defined (they can be negative) because the chicken is travelling at a constant velocity. Since β_1 is restricting the y-velocity of the chicken to be strictly positive, it follows that $t_1 < t_2$.

Also note that s_1 is equal to the time it takes for the front of the car to attain an x-value equal to that of the chicken if the chicken were to continue travelling at the given velocity forever; similarly, s_2 is equal to the time it takes for the back of the car to attain an x-value equal to that of the chicken if the chicken were to continue travelling at the given velocity. Note that s_1 and s_2 are unique and well defined because the x-velocities of the chicken and car are different and because the car and chicken travel at constant velocities. Note that since we assumed that $l = 0$, it follows that $s_1 = s_2$.

Then it is easy to see that that path of the chicken intersects with the path of the car if and only if the interval $[t_1, t_2]$ intersects with the interval $[s_1, s_2]$ (this is still true without assuming that $l = 0$). Furthermore, it is just as easy to see that the chicken collides with the car at all times in $[t_1, t_2] \cap [s_1, s_2]$. Hence, finally, the chicken only intersects with the car if any times of intersection occur after T, or $\min s_1, t_1 > T$.

The tricky part is that this is still true even when cars are being teleported back to the beginning, *because it is assumed that* $l = 0$, which effectively makes it so that no more than one car can exist in the same path within a stretch of L units of road; otherwise, we would also have to account for the intersection of the parts of the cars that just teleported (which model the incoming new car), which makes the arithmetic significantly more complicated.

Thus, if $v_x \neq u_x$, the condition that ensures that the chicken will not collide with the car in the next T units of time is given by

$$t_2 < s_1 \lor s_2 < t_1 \lor t_1 > T \lor s_1 > T$$

if $s_1 < s_2$ and

$$t_2 < s_2 \lor s_1 < t_1 \lor t_1 > T \lor s_2 > T$$

otherwise.

Finally, in ϵ, we let physics evolve, allowing the chicken and car to travel according to their differential equations $c_x' = u_x$ (the car only travels horizontally in the x-direction) and $\mathbf{k}' = \mathbf{v}'$, respectively, for a period time at most T.

With this intuition, we can prove the safety of the program. Given initial conditions $A \land B$ and safety condition S, this amounts to proving the formula

$$A \land B \to [\chi^*]S \tag{1}$$

which logically states that given initial conditions A and B (the chicken starts at the origin with zero velocity and the car starts within the road with some positive x-velocity u_x), any run of the program χ^*

will result in S being true (meaning the chicken does not intersect with the car in 3-dimensional space).

The proof of formula 1 was automatically done by Keymaera and is given in the Keymaera file `chick_1car.key.proof`. Because of this, and since the intuitive explanation of χ^* basically gives a proof sketch, instead of giving the full-fledged proof here, I instead illustrate the direction of the proof and the techniques used in the proof.

The program begins with the repetition rule $*$, for which I use an invariant to apply the induction rule. The invariant I use is $I \equiv S \wedge B \wedge c_z = 0$, which state the conditions on any constants used plus the fact that the car stays on the ground and that the chicken should never have collided with the car to start off with. Going through the first parts of the proof:

$$\rightarrow_l,\ ind\ \frac{\mathbb{R}\ \dfrac{*}{A, B \vdash I = 0} \qquad ax\ \dfrac{*}{I \vdash S} \qquad X_1}{A \wedge B \rightarrow [\chi^*]S}$$

Continuing on with branch X_1:

$$\cdots\ \frac{\ldots}{I \vdash \{t := 0\}[\alpha \cup \beta][\epsilon]I}$$

From here on, straightforward applications of the basic sequent rules and ODE solves finishes the proof.

4.2 Distributed Model

The proof for the safety of the Distributed Model utilized much rules in *QdL* such as quantified differential cuts and quantified differential invariants.

We use the same notation as before, but extend that for the cars as follows. For each car $i \in C$ (i.e., for each object i of sort C), let $\mathbf{c}(i)$ denote the position of i, and u_x denote the x-velocity of i. For this simplified version, we will also have that the width w, length l and height h of every car is the same. Further, to express that fact that the cars are all in one lane, we will have $c_y(i) = c_y(j) \wedge u_x = u_x(j)$ for every $i \neq j \in C$. For simplicity, we will just write u_x for the velocity of any one car. Finally, there is some gap between the cars, which we express by $c_x(i) - c_x(j) < -l \vee c_x(i) - c_x(j) > l$ for cars $i \neq j \in C$.

Now the quantified hybrid program for the distributed model is given by χ^*:

$$\begin{aligned}
A &\equiv \mathbf{k} = 0 \wedge \mathbf{v} = 0 \\
B_1 &\equiv \forall i : C.\ 0 < c_y(i) - \frac{w}{2} \wedge c_y(i) + \frac{w}{2} < H \\
B_2 &\equiv \forall i : C.\ c_z(i) = 0 \\
B &\equiv B_1 \wedge B_2 \wedge u_x > 0 \\
C &\equiv H > 0 \wedge w > 0 \wedge h \geq 0 \wedge l \geq 0 \wedge T > 0 \\
\alpha &\equiv k_z := h + 1; v_x := 0; v_y := 0; v_z := 0 \\
\beta_1 &\equiv k_z := 0; v_x := *; v_y := *; v_z := 0; ?(v_y > 0) \\
Q_1 &\equiv \forall i : C.\ k_x < c_x(i) - l \vee c_x(i) < k_x \vee t_1(i) > T \\
\beta_2 &\equiv \gamma_1; \gamma_2; \mathsf{if}\ v_x = u_x\ \mathsf{then}\ ?Q_1\ \mathsf{else}\ \gamma_3; \gamma_4; \delta_1; \delta_2\ \mathsf{fi} \\
\beta &\equiv \beta_1; \beta_2 \\
\gamma_1 &\equiv \forall i : C.\ t_1(i) := \frac{c_y(i) - \frac{w}{2} - k_y}{v_y} \\
\gamma_2 &\equiv \forall i : C.\ t_2(i) := \frac{c_y(i) + \frac{w}{2} - k_y}{v_y} \\
\gamma_3 &\equiv \forall i : C.\ s_1(i) := \frac{c_x(i) - k_x}{v_x - u_x} \\
\gamma_4 &\equiv \forall i : C.\ s_2(i) := \frac{c_x(i) - l - k_x}{v_x - u_x} \\
\delta_1 &\equiv ?\,(\forall i : C.\ s_1(i) \leq s_2(i) \rightarrow (Q_2(i))) \\
Q_2(i) &\equiv t_2(i) < s_1(i) \vee s_2(i) < t_1(i) \vee t_1(i) > T \vee s_1(i) > T \\
Q_3(i) &\equiv t_2(i) < s_2(i) \vee s_1(i) < t_1(i) \vee t_1(i) > T \vee s_2(i) > T \\
\delta_2 &\equiv ?\,(\forall i : C.\ s_1(i) > s_2(i) \rightarrow (Q_3(i))) \\
\epsilon &\equiv \forall i : C.\ \{c_x(i)' = u_x, \mathbf{k}' = \mathbf{v}, t' = 1, t \leq T\} \\
\chi &\equiv t := 0; (\alpha \cup \beta); \epsilon
\end{aligned}$$

Then the hybrid program for the Distributed Model is $\mu_D \equiv \chi^*$. The safety condition for this program is

$$\begin{aligned}
S = \forall i : C(\neg(0 \leq c_x(i) - k_x \leq l \\
\wedge |c_y(i) - k_y| \leq \frac{w}{2} \wedge 0 \leq k_z - c_z(i) \leq h)),
\end{aligned}$$

and the initial conditions are $A \wedge B \wedge C$.

The analysis is just the same as the Teleportation Model, except that we no longer have to account for teleportation of cars in calculating path intersections between the chicken and the cars, which makes the overall general arithmetic much simpler. However, the distributed nature still gives the power of the ability to model many cars at once, because the cars all behave very similarly. Since the overall analysis is the same, we omit the intuitive analysis.

With the same intuition as before, we can prove the safety of the program. Given initial conditions

$A \wedge B \wedge C$ and safety condition S, this amounts to proving the formula

$$A \wedge B \wedge C \rightarrow [\chi^*]S \tag{2}$$

which logically states that given initial conditions A and B and C (the chicken starts at the origin with zero velocity and the car starts within the road with some positive x-velocity u_x), any run of the program χ^* will result in S being true (meaning the chicken does not intersect with the car in 3-dimensional space).

The proof of formula 1 was automatically done by Keymaera and is given in the Keymaera file `chick_1lane.key.proof`. As before, since the intuitive explanation of χ^* basically gives a proof sketch, instead of giving the full-fledged proof here, I instead illustrate the direction of the proof and the techniques used in the proof.

One of the biggest new techniques that was utilized in this proof was the quantified differential cut (QDC) and the quantified differential invariant (QDI). These rules are really no different from the normal differential cut and the normal differential invariant, except for the fact that properties that are being inducted on or cut in can apply to all objects of the same sort, which makes it a powerful tool for our proof.

As before, rhe program begins with the repetition rule $*$, for which I use an invariant to apply the induction rule. The invariant I use is $I \equiv S \wedge B \wedge C$, which state the conditions on any constants used plus the fact that all cars stay on the ground and that the chicken should not be colliding with any car to start off with. Going through the first parts of the proof:

$$\rightarrow_l,\ ind\ \frac{\mathbb{R}\ \dfrac{*}{A, B \vdash I = 0} \qquad ax\ \dfrac{*}{I \vdash S} \qquad Y_1}{A \wedge B \rightarrow [\chi^*]S}$$

This time, I will partially complete the sequent proof in order to demonstrate applications of some of the QdL sequent rules.

Let is continue with branch Y_1. Let $X' = \Theta$ denote the differential equations $c_x(i)' = u_x, \mathbf{k}' = \mathbf{v}, t' = 1$. Then we apply a quantified differential cut, cutting in C.

$$\dfrac{QDC\ \dfrac{Y_2 \qquad Y_3}{I \vdash \{t := 0; \alpha\}[\forall i : C.\ \{X' = \Theta, t \leq T]I}}{[\cup], [:=]\ \dfrac{\ldots}{I \vdash \{t := 0\}[\alpha \cup \beta][\forall i : C.\ \{X' = \Theta, t \leq T]I}}$$

From here on, straightforward applications of the basic sequent rules and ODE solves finishes the proof. Continuing on with branch Y_2: Applying quantified differential invariant here gives

$$QDI\ \frac{ax\ \dfrac{*}{I \vdash C} \qquad (')\ sub\ \dfrac{\mathbb{R}\ \dfrac{*}{\vdash 0 = 0 \geq 0}}{\vdash (C')^{\Theta}_{X'}}}{I \vdash \{t := 0\}[\alpha \cup \beta][\forall i : C.\ \{X' = \Theta, t \leq T]C}$$

Continuing on with branch Y_3: Doing a similar sequence of steps will close this branch as well. By first applying a quantified differential cut with cut formula K and then applying quantified differential invariant, where

$$\begin{aligned} K \equiv \forall i : C(c_x(i) = c_{x_0}(i) + u * t) \\ \wedge \forall i : C(c_y(i) = c_{y_0}(i)) \\ \wedge \forall i : C(c_z(i) = 0) \\ \wedge k_x = k_{x_0} + v_x * t \\ \wedge k_y = k_{y_0} + v_y * t \\ \wedge k_z = h + 1 \wedge t >= 0 \end{aligned}$$

and where $c_{x_0}, c_{y_0}, k_{x_0}$ and k_{y_0} are auxiliary variables that store the values of the positions of the chicken and the car before the differential equations evolve, this branch can close after basic sequent proof rules and quantifier elimination.

For details, the reader may inspect the proof file `chick_1lane.key.proof` for the corresponding Keymaera file `chick_1lane.key`.

5 Hybrid Programs and Existence Proofs

In the previous sections, we were able to successfully prove that in both the simpler Teleportation Model and the more general Distributed Model, the chicken would always be safe. However, it remains to show that in either model, there does exist a run of the hybrid program such that the chicken *can* indeed make

it to the other side of the road. This is crucial since it will show that not only the choices the chicken has are safe, but that it is possible for it to safely *cross* the road as well.

Because the Distributed Model is more general, we only prove existence for the Distributed Model and leave it to the reader to convice himself that solutions do exist for the Teleportation Model as well.

Due to the length of the proofs, the proofs and discussions following the proofs have been relegated to the Appendices.

6 Conclusion

The original intention of this project was to model the system of a chicken crossing a road as realistically as possible in order to prove that the chicken could cross the road. The ideal program would have been a time-triggered controller that allowed the chicken to choose a bounded acceleration in any direction. There would be preconditions set up on the initial velocity and position of the cars to always guarantee a path given the reaction time and acceleration capabilities of the chicken. Fortunately, we were able to prove a simplified version for any number cars with the help of quantified differential logic.

What we ended up with was a time-triggered controller that allowed the chicken to choose a velocity in any direction, assuming instantaneous acceleration. We had an event-based controller that used proper preconditions to guarantee safety, but we were forced to include a shortcut in the form of a slightly unrealistic action to guarantee safety in a time-based controller. Initially, we could only prove our programs for cases where there were at most 2 cars. The main reason for our shortcomings was due to the complexity of our initial goals. We had trouble imaging what type of preconditions were necessary to guarantee a safe option at each time-trigger. The few ideas we had soon failed as we tried convert them into mathematical formulas. We did not use acceleration in the chicken's motion because of the drastic increase in complexity it introduced. Adding acceleration into the mix would introduce another layer to our differential equations and increase the mathematical complexity of all existing formulas by a great deal. Furthermore, we strongly believe that in the case of a chicken moving with relatively slow speed, instantaneous changes in velocity is not an overly inaccurate estimate of the real world.

Coming up with the distributed model using QdL was a goal we had not planned on achieving. We imagined that we could use our teleportation tactic to simulate an infinite stream of cars of any shape. However, as we began trying to prove programs with longer cars, we realized that prior methods would no longer work, and simply adding another car to the picture to simulate multiple lanes took exponential amount of effort to prove. We had to turn to other methods, and the idea of QdL fit perfectly. Having the ability to represent a car as a class allowed us to simulate the infinite stream of cars by proving general properties of that class.

The most important thing we learned while working on this project was how to convert a real world cyber physical system into a dL hybrid program by making the right simplifications. We were able to make progress only if we took the time to brainstorm and design the most complete models. This experience will prepare us for the next time we want to model real world problems.

Each party member did comparable amounts of work.

References

[1] Andre Platzer. *A Complete Axiomatization of Quantified Differential Dynamic Logic for Distributed Hybrid Systems*. 2012.

[2] Andre Platzer. *Quantified Differential Dynamic Logic for Distributed Hybrid Systems*. In Anuj Dawar and Helmut Veith, editors, CSL, volume 6247 of LNCS, pages 469-483. Springer, 2010.

7 Appendix A

In this section, we present proof of the existence for certain, realistic cases of the Distributed Model. We do this by first proving a much simpler case, and then slowly building it back up to the general Distributed Model case.

7.1 One Car

Here, we prove that in the Distributed Model for only one car (basically, the road is infinitely long and there is only one car), the chicken has a strategy to cross the road.

We use the same notation as before. Let q be some auxiliary variable.

We introduce some abbreviations:

$$A \equiv k_x = 0 \land k_y = 0 \land k_z = 0 \land v_x = 0 \land v_y = 0 \land v_z = 0 \land 0 < c_y - \frac{w}{2} \land c_y + \frac{w}{2} < H \land c_z = 0 \land u_x > 0$$

$$B \equiv H > 0 \land w > 0 \land h \geq 0 \land l \geq 0 \land T > 0$$

$$\alpha \equiv k_z := h + 1; v_x := 0; v_y := 0; v_z := 0$$

$$\beta \equiv k_z := 0; v_x := *; v_y := *; v_z := 0; ?(v_y > 0); \gamma_1; \gamma_2; \text{if } v_x = u_x \text{ then } ?(k_x < c_x - l \lor c_x < k_x \lor t_1 > T) \text{ else } \gamma_3; \gamma_4; \delta_1; \delta_2 \text{ fi}$$

$$\gamma_1 \equiv t_1 := \frac{c_y - \frac{w}{2} - k_y}{v_y}$$

$$\gamma_2 \equiv t_2 := \frac{c_y + \frac{w}{2} - k_y}{v_y}$$

$$\gamma_3 \equiv s_1 := \frac{c_x - k_x}{v_x - u_x}$$

$$\gamma_4 \equiv s_2 := \frac{c_x - l - k_x}{v_x - u_x}$$

$$\delta_1 \equiv \text{if } s_1 > s_2 \text{ then } q := s_1; s_1 := s_2; s_2 := q \text{ fi}$$

$$\delta_2 \equiv ?\,(t_2 < s_1 \lor s_2 < t_1 \lor t_1 > T \lor s_1 > T)$$

$$\epsilon \equiv \{c_x' = u_x, k_x' = v_x, k_y' = v_y, k_z' = v_z, t' = 1, t \leq T\}$$

$$\chi \equiv t := 0; (\alpha \cup \beta); \epsilon$$

Then we wish to prove the following:

$$A \land B \to \langle \chi^* \rangle\, k_y \geq H$$

This is the same as the program found in the key file `chick_1car_ext.key`. We will prove this in multiple steps:

Let $p(\mathbf{k}, \mathbf{c})$ be the predicate $\langle \chi^* \rangle\, k_y \geq H$, where the vectors in the arguments have their obvious meanings.

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\cdots \quad \dfrac{Y_1}{\forall \mathbf{k}, \mathbf{c}\,((k_y \geq H \vee \{t := 0\}\,\langle\alpha\rangle\,\langle\epsilon\rangle\,p(\mathbf{k},\mathbf{c}) \vee \{t := 0\}\,\langle\beta\rangle\,\langle\epsilon\rangle\,p(\mathbf{k},\mathbf{c})) \to p(\mathbf{k},\mathbf{c})) \to (A \wedge B \to p(\mathbf{k},\mathbf{c}))}}{\forall \mathbf{k}, \mathbf{c}\,((k_y \geq H \vee \langle\chi\rangle\,p(\mathbf{k},\mathbf{c})) \to p(\mathbf{k},\mathbf{c})) \to (A \wedge B \to p(\mathbf{k},\mathbf{c}))}\,\langle;\rangle,\,\langle:=\rangle,\,\langle\cup\rangle}{\forall \mathbf{k}, \mathbf{c}\,((k_y \geq H \vee \langle\chi\rangle\,\langle\chi^*\rangle\,k_y \geq H) \to \langle\chi^*\rangle\,k_y \geq H) \to (A \wedge B \to \langle\chi^*\rangle\,k_y \geq H)}\,US}{A \wedge B \to \langle\chi^*\rangle\,k_y \geq H}\,\langle*\rangle,\,\forall,\,MP$$

Let

$$X_1 \equiv \{t := 0\}\,\langle\alpha\rangle\,\langle\epsilon\rangle\,p(\mathbf{k},\mathbf{c})$$
$$X_2 \equiv \{t := 0\}\,\langle\beta\rangle\,\langle\epsilon\rangle\,p(\mathbf{k},\mathbf{c})$$

Reduction of X_1:

Let

$$\zeta_1 \equiv c_x := c_x + u_x \cdot \tilde{r}; k_x := k_x + v_x \cdot \tilde{r}; k_y := k_y + v_y \cdot \tilde{r}; k_z := k_z + v_z \cdot \tilde{r}; t := t + \tilde{r}$$
$$\zeta_2 \equiv c_x := c_x + u_x \cdot r; k_x := k_x + v_x \cdot r; k_y := k_y + v_y \cdot r; k_z := k_z + v_z \cdot r; t := t + r$$

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\mathbb{R}\,\dfrac{\exists 0 \leq r \leq T.\ p((k_x,k_y,h+1),(c_x+u_x\cdot r,c_y,c_z))}{\exists r \geq 0\,((\forall 0 \leq \tilde{r} \leq r\ \tilde{r} \leq T) \wedge p((k_x,k_y,h+1),(c_x+u_x\cdot r,c_y,c_z)))}}{\{t := 0; k_z := h+1; v_x := 0; v_y := 0; v_z := 0\}\exists r \geq 0\,((\forall 0 \leq \tilde{r} \leq r\,\langle\zeta_1\rangle\,t \leq T) \wedge \langle\zeta_2\rangle\,p(\mathbf{k},\mathbf{c}))}\,subst}{\{t := 0; k_z := h+1; v_x := 0; v_y := 0; v_z := 0\}\,\langle\epsilon\rangle\,p(\mathbf{k},\mathbf{c})}\,\langle'\rangle}{\{t := 0\}\,\langle\alpha\rangle\,\langle\epsilon\rangle\,p(\mathbf{k},\mathbf{c})}\,\langle;\rangle,\,\langle:=\rangle$$

Reduction of X_2:

Let

$$\zeta_3 \equiv t := 0; k_z := 0; v_x := \widetilde{v_x}; v_y := \widetilde{v_y}; v_z := 0$$
$$\zeta_4 \equiv t := 0; k_z := 0; v_x := \widetilde{v_x}; v_y := \widetilde{v_y}; v_z := 0; t_1 := \frac{c_y - \frac{w}{2} - k_y}{\widetilde{v_y}}; t_2 := \frac{c_y + \frac{w}{2} - k_y}{\widetilde{v_y}}$$
$$\zeta_5 \equiv ?(k_x < c_x - l \vee c_x < k_x \vee t_1 > T)$$

$$\langle;\rangle,\ \langle:=\rangle,\ \langle:*\rangle\ \frac{\langle?\rangle,\ \langle:=\rangle,\ sub\ \dfrac{\textit{if, sub}\ \dfrac{\exists\vee\ \dfrac{Z_1\vee Z_2}{\exists\widetilde{v_x},\widetilde{v_y}.\ \widetilde{v_y}>0\wedge((\widetilde{v_x}=u_x\wedge\{\zeta_4\}\langle\zeta_5\rangle\langle\epsilon\rangle p(\mathbf{k},\mathbf{c}))\vee(\widetilde{v_x}\neq u_x\wedge\{\zeta_4\}\langle\gamma_3;\gamma_4;\delta_1;\delta_2\rangle\langle\epsilon\rangle p(\mathbf{k},\mathbf{c})))}}{\exists\widetilde{v_x},\widetilde{v_y}.\ \widetilde{v_y}>0\wedge\{\zeta_4\}\langle\textsf{if }v_x=u_x\textsf{ then }\zeta_5\textsf{ else }\gamma_3;\gamma_4;\delta_1;\delta_2\textsf{ fi}\rangle\langle\epsilon\rangle p(\mathbf{k},\mathbf{c})}}{\exists\widetilde{v_x},\widetilde{v_y}.\ \{\zeta_3\}\langle?(v_y>0)\rangle\langle\gamma_1\rangle\langle\gamma_2\rangle\langle\textsf{if }v_x=u_x\textsf{ then }\zeta_5\textsf{ else }\gamma_3;\gamma_4;\delta_1;\delta_2\textsf{ fi}\rangle\langle\epsilon\rangle p(\mathbf{k},\mathbf{c})}}{\{t:=0\}\langle\beta\rangle\langle\epsilon\rangle p(\mathbf{k},\mathbf{c})}$$

where $\widetilde{v_x}$ and $\widetilde{v_y}$ are fresh, and where

$$Z_1\equiv\exists\widetilde{v_x},\widetilde{v_y}.\ \widetilde{v_y}>0\wedge\widetilde{v_x}=u_x\wedge\{\zeta_4\}\langle\zeta_5\rangle\langle\epsilon\rangle p(\mathbf{k},\mathbf{c})$$
$$Z_2\equiv\exists\widetilde{v_x},\widetilde{v_y}.\ \widetilde{v_y}>0\wedge\widetilde{v_x}\neq u_x\wedge\{\zeta_4\}\langle\gamma_3;\gamma_4;\delta_1;\delta_2\rangle\langle\epsilon\rangle p(\mathbf{k},\mathbf{c})$$

Reduction of Z_1:

Let

$$F_1\equiv\widetilde{v_y}>0\wedge\widetilde{v_x}=u_x\wedge\left(k_x<c_x-l\vee c_x<k_x\vee\frac{c_y-\frac{w}{2}-k_y}{\widetilde{v_y}}>T\right)$$

$$\langle;\rangle,\ \langle:=\rangle,\ \langle:*\rangle\ \frac{\langle?\rangle,\ \langle:=\rangle,\ sub\ \dfrac{\langle'\rangle,\ sub,\ \mathbb{R}\ \dfrac{ax\ \dfrac{\exists\widetilde{v_x},\widetilde{v_y},0\leq r\leq T.\ (F_1\wedge p((k_x+\widetilde{v_x}\cdot r,k_y+\widetilde{v_y}\cdot r,0),(c_x+u_x\cdot r,c_y,c_z)))}{\exists\widetilde{v_x},\widetilde{v_y}.\ (F_1\wedge\exists 0\leq r\leq T.\ p((k_x+\widetilde{v_x}\cdot r,k_y+\widetilde{v_y}\cdot r,0),(c_x+u_x\cdot r,c_y,c_z)))}}{\exists\widetilde{v_x},\widetilde{v_y}.\ \widetilde{v_y}>0\wedge\widetilde{v_x}=u_x\wedge\left(k_x<c_x-l\vee c_x<k_x\vee\frac{c_y-\frac{w}{2}-k_y}{\widetilde{v_y}}>T\right)\wedge\{\zeta_4\}\langle\epsilon\rangle p(\mathbf{k},\mathbf{c})}}{\exists\widetilde{v_x},\widetilde{v_y}.\ \widetilde{v_y}>0\wedge\widetilde{v_x}=u_x\wedge\{\zeta_4\}\langle\zeta_5\rangle\langle\epsilon\rangle p(\mathbf{k},\mathbf{c})}$$

Reduction of Z_2:

Let

$$\zeta_6\equiv t:=0;k_z:=0;v_x:=\widetilde{v_x};v_y:=\widetilde{v_y};v_z:=0;t_1:=\frac{c_y-\frac{w}{2}-k_y}{\widetilde{v_y}};t_2:=\frac{c_y+\frac{w}{2}-k_y}{\widetilde{v_y}};s_1:=\frac{c_x-k_x}{\widetilde{v_x}-u_x};s_2:=\frac{c_x-l-k_x}{\widetilde{v_x}-u_x}$$

$$\langle;\rangle,\ \langle:=\rangle,\ sub\ \frac{\textit{if, sub, }\vee\ \dfrac{Z_3\vee Z_4}{\exists\widetilde{v_x},\widetilde{v_y}.\ \widetilde{v_y}>0\wedge\widetilde{v_x}\neq u_x\wedge\{\zeta_6\}\langle\delta_1\rangle\langle\delta_2\rangle\langle\epsilon\rangle p(\mathbf{k},\mathbf{c})}}{\exists\widetilde{v_x},\widetilde{v_y}.\ \widetilde{v_y}>0\wedge\widetilde{v_x}\neq u_x\wedge\{\zeta_4\}\langle\gamma_3;\gamma_4;\delta_1;\delta_2\rangle\langle\epsilon\rangle p(\mathbf{k},\mathbf{c})}$$

where

$$Z_3 \equiv \exists \widetilde{v_x}, \widetilde{v_y}.\ \widetilde{v_y} > 0 \wedge \widetilde{v_x} \neq u_x \wedge \left(\frac{c_x - k_x}{\widetilde{v_x} - u_x} > \frac{c_x - l - k_x}{\widetilde{v_x} - u_x}\right) \wedge \{\zeta_6\}\, \langle q := s_1; s_1 := s_2; s_2 := q\rangle\, \langle \delta_2\rangle\, \langle \epsilon\rangle\, p(\mathbf{k}, \mathbf{c})$$

$$Z_4 \equiv \exists \widetilde{v_x}, \widetilde{v_y}.\ \widetilde{v_y} > 0 \wedge \widetilde{v_x} \neq u_x \wedge \left(\frac{c_x - k_x}{\widetilde{v_x} - u_x} \leq \frac{c_x - l - k_x}{\widetilde{v_x} - u_x}\right) \wedge \{\zeta_6\}\, \langle \delta_2\rangle\, \langle \epsilon\rangle\, p(\mathbf{k}, \mathbf{c})$$

Reduction of Z_3:

Let

$$\zeta_7 \equiv t := 0; k_z := 0; v_x := \widetilde{v_x}; v_y := \widetilde{v_y}; v_z := 0; t_1 := \frac{c_y - \frac{w}{2} - k_y}{\widetilde{v_y}}; t_2 := \frac{c_y + \frac{w}{2} - k_y}{\widetilde{v_y}}; s_1 := \frac{c_x - l - k_x}{\widetilde{v_x} - u_x}; s_2 := \frac{c_x - k_x}{\widetilde{v_x} - u_x}; q := \frac{c_x - k_x}{\widetilde{v_x} - u_x}$$

$$F_2 \equiv \frac{c_x - k_x}{\widetilde{v_x} - u_x} > \frac{c_x - l - k_x}{\widetilde{v_x} - u_x}$$

$$F_3 \equiv \left(\frac{c_y + \frac{w}{2} - k_y}{\widetilde{v_y}} < \frac{c_x - l - k_x}{\widetilde{v_x} - u_x} \vee \frac{c_x - k_x}{\widetilde{v_x} - u_x} < \frac{c_y - \frac{w}{2} - k_y}{\widetilde{v_y}} \vee \frac{c_y - \frac{w}{2} - k_y}{\widetilde{v_y}} > T \vee \frac{c_x - l - k_x}{\widetilde{v_x} - u_x} > T\right)$$

$$\langle;\rangle,\ \langle:=\rangle,\ \langle ?\rangle,\ sub\ \frac{\langle'\rangle,\ sub,\ \mathbb{R}\ \dfrac{ax\ \dfrac{\exists \widetilde{v_x}, \widetilde{v_y}, 0 \leq r \leq T.\ \widetilde{v_y} > 0 \wedge \widetilde{v_x} \neq u_x \wedge F_2 \wedge F_3 \wedge p((k_x + \widetilde{v_x} \cdot r, k_y + \widetilde{v_y} \cdot r, 0), (c_x + u_x \cdot r, c_y, c_z))}{\exists \widetilde{v_x}, \widetilde{v_y}.\ \widetilde{v_y} > 0 \wedge \widetilde{v_x} \neq u_x \wedge F_2 \wedge F_3 \wedge \exists 0 \leq r \leq T.\ p((k_x + \widetilde{v_x} \cdot r, k_y + \widetilde{v_y} \cdot r, 0), (c_x + u_x \cdot r, c_y, c_z))}}{\exists \widetilde{v_x}, \widetilde{v_y}.\ \widetilde{v_y} > 0 \wedge \widetilde{v_x} \neq u_x \wedge F_2 \wedge F_3 \wedge \{\zeta_7\}\, \langle \epsilon\rangle\, p(\mathbf{k}, \mathbf{c})}}{\exists \widetilde{v_x}, \widetilde{v_y}.\ \widetilde{v_y} > 0 \wedge \widetilde{v_x} \neq u_x \wedge F_2 \wedge \{\zeta_6\}\, \langle q := s_1; s_1 := s_2; s_2 := q\rangle\, \langle \delta_2\rangle\, \langle \epsilon\rangle\, p(\mathbf{k}, \mathbf{c})}$$

Reduction of Z_4:

Let

$$F_4 \equiv \left(\frac{c_y + \frac{w}{2} - k_y}{\widetilde{v_y}} < \frac{c_x - k_x}{\widetilde{v_x} - u_x} \vee \frac{c_x - l - k_x}{\widetilde{v_x} - u_x} < \frac{c_y - \frac{w}{2} - k_y}{\widetilde{v_y}} \vee \frac{c_y - \frac{w}{2} - k_y}{\widetilde{v_y}} > T \vee \frac{c_x - k_x}{\widetilde{v_x} - u_x} > T\right)$$

$$\langle ?\rangle,\ sub\ \frac{\langle'\rangle,\ sub,\ \mathbb{R}\ \dfrac{ax\ \dfrac{\exists \widetilde{v_x}, \widetilde{v_y}, 0 \leq r \leq T.\ \widetilde{v_y} > 0 \wedge \widetilde{v_x} \neq u_x \wedge \neg F_2 \wedge F_4 \wedge p((k_x + \widetilde{v_x} \cdot r, k_y + \widetilde{v_y} \cdot r, 0), (c_x + u_x \cdot r, c_y, c_z))}{\exists \widetilde{v_x}, \widetilde{v_y}.\ \widetilde{v_y} > 0 \wedge \widetilde{v_x} \neq u_x \wedge \neg F_2 \wedge F_4 \wedge \exists 0 \leq r \leq T.\ p((k_x + \widetilde{v_x} \cdot r, k_y + \widetilde{v_y} \cdot r, 0), (c_x + u_x \cdot r, c_y, c_z))}}{\exists \widetilde{v_x}, \widetilde{v_y}.\ \widetilde{v_y} > 0 \wedge \widetilde{v_x} \neq u_x \wedge \neg F_2 \wedge F_4 \wedge \{\zeta_6\}\, \langle \epsilon\rangle\, p(\mathbf{k}, \mathbf{c})}}{\exists \widetilde{v_x}, \widetilde{v_y}.\ \widetilde{v_y} > 0 \wedge \widetilde{v_x} \neq u_x \wedge \left(\frac{c_x - k_x}{\widetilde{v_x} - u_x} \leq \frac{c_x - l - k_x}{\widetilde{v_x} - u_x}\right) \wedge \{\zeta_6\}\, \langle \delta_2\rangle\, \langle \epsilon\rangle\, p(\mathbf{k}, \mathbf{c})}$$

Proof of Y_1:

Let

$$
\begin{aligned}
W_1 &\equiv \exists 0 \leq r \leq T.\ p((k_x, k_y, h+1), (c_x + u_x \cdot r, c_y, c_z)) \\
W_2 &\equiv \exists \widetilde{v_x}, \widetilde{v_y}, 0 \leq r \leq T.\ (F_1 \wedge p((k_x + \widetilde{v_x} \cdot r, k_y + \widetilde{v_y} \cdot r, 0), (c_x + u_x \cdot r, c_y, c_z))) \\
W_3 &\equiv \exists \widetilde{v_x}, \widetilde{v_y}, 0 \leq r \leq T.\ \widetilde{v_y} > 0 \wedge \widetilde{v_x} \neq u_x \wedge F_2 \wedge F_3 \wedge p((k_x + \widetilde{v_x} \cdot r, k_y + \widetilde{v_y} \cdot r, 0), (c_x + u_x \cdot r, c_y, c_z)) \\
W_4 &\equiv \exists \widetilde{v_x}, \widetilde{v_y}, 0 \leq r \leq T.\ \widetilde{v_y} > 0 \wedge \widetilde{v_x} \neq u_x \wedge \neg F_2 \wedge F_4 \wedge p((k_x + \widetilde{v_x} \cdot r, k_y + \widetilde{v_y} \cdot r, 0), (c_x + u_x \cdot r, c_y, c_z))
\end{aligned}
$$

Now, we can continue on with the first proof:

$$
\frac{\to \dfrac{\forall \mathbf{k}, \mathbf{c}\,((k_y \geq H \vee W_1 \vee W_2 \vee W_3 \vee W_4) \to p(\mathbf{k}, \mathbf{c})) \to (\neg(A \wedge B) \vee (A \wedge B \wedge p(\mathbf{k}, \mathbf{c})))}{\forall \mathbf{k}, \mathbf{c}\,((k_y \geq H \vee W_1 \vee W_2 \vee W_3 \vee W_4) \to p(\mathbf{k}, \mathbf{c})) \to (A \wedge B \to p(\mathbf{k}, \mathbf{c}))}}{\forall \mathbf{k}, \mathbf{c}\,((k_y \geq H \vee \{t := 0\}\,\langle\alpha\rangle\,\langle\epsilon\rangle\,p(\mathbf{k}, \mathbf{c}) \vee \{t := 0\}\,\langle\beta\rangle\,\langle\epsilon\rangle\,p(\mathbf{k}, \mathbf{c})) \to p(\mathbf{k}, \mathbf{c})) \to (A \wedge B \to p(\mathbf{k}, \mathbf{c}))}
$$

As a reminder, we have

$$
\begin{aligned}
A &\equiv k_x = 0 \wedge k_y = 0 \wedge k_z = 0 \wedge v_x = 0 \wedge v_y = 0 \wedge v_z = 0 \wedge 0 < c_y - \frac{w}{2} \wedge c_y + \frac{w}{2} < H \wedge c_z = 0 \wedge u_x > 0 \\
B &\equiv H > 0 \wedge w > 0 \wedge h \geq 0 \wedge l \geq 0 \wedge T > 0 \\
F_1 &\equiv \widetilde{v_y} > 0 \wedge \widetilde{v_x} = u_x \wedge \left(k_x < c_x - l \vee c_x < k_x \vee \frac{c_y - \frac{w}{2} - k_y}{\widetilde{v_y}} > T\right) \\
F_2 &\equiv \frac{c_x - k_x}{\widetilde{v_x} - u_x} > \frac{c_x - l - k_x}{\widetilde{v_x} - u_x} \\
F_3 &\equiv \left(\frac{c_y + \frac{w}{2} - k_y}{\widetilde{v_y}} < \frac{c_x - l - k_x}{\widetilde{v_x} - u_x} \vee \frac{c_x - k_x}{\widetilde{v_x} - u_x} < \frac{c_y - \frac{w}{2} - k_y}{\widetilde{v_y}} \vee \frac{c_y - \frac{w}{2} - k_y}{\widetilde{v_y}} > T \vee \frac{c_x - l - k_x}{\widetilde{v_x} - u_x} > T\right) \\
F_4 &\equiv \left(\frac{c_y + \frac{w}{2} - k_y}{\widetilde{v_y}} < \frac{c_x - k_x}{\widetilde{v_x} - u_x} \vee \frac{c_x - l - k_x}{\widetilde{v_x} - u_x} < \frac{c_y - \frac{w}{2} - k_y}{\widetilde{v_y}} \vee \frac{c_y - \frac{w}{2} - k_y}{\widetilde{v_y}} > T \vee \frac{c_x - k_x}{\widetilde{v_x} - u_x} > T\right)
\end{aligned}
$$

If the left hand side $\forall \mathbf{k}, \mathbf{c}\,((k_y \geq H \vee W_1 \vee W_2 \vee W_3 \vee W_4) \to p(\mathbf{k}, \mathbf{c}))$ implies $\neg(A \wedge B)$, then there is nothing to show, so we may instead simply prove

$$A \wedge B \wedge \forall \mathbf{k}, \mathbf{c}\,((k_y \geq H \vee W_1 \vee W_2 \vee W_3 \vee W_4) \to p(\mathbf{k}, \mathbf{c})) \to p(\mathbf{k}, \mathbf{c})$$

Then the proof continues as:

$$
eq\ \frac{A \wedge B \wedge \forall \mathbf{k}, \mathbf{c}\,((k_y \geq H \vee W_1 \vee W_2 \vee W_3 \vee W_4) \to p(\mathbf{k}, \mathbf{c})) \to p((0,0,0), (c_x, c_y, 0))}{A \wedge B \wedge \forall \mathbf{k}, \mathbf{c}\,((k_y \geq H \vee W_1 \vee W_2 \vee W_3 \vee W_4) \to p(\mathbf{k}, \mathbf{c})) \to p(\mathbf{k}, \mathbf{c})}
$$

Now, we can consider the W_i's as "rules" that decide which values of $\mathbf{k}$ and $\mathbf{c}$ makes $p(\mathbf{k}, \mathbf{c})$ true, and prove efficiency by working backwards. Initially, the chicken starts at $(0, 0, 0)$ and the car at $(c_x, c_y, 0)$. It is not too hard to see that the chicken can choose to hover in place until

the car passes it (using rule W_1 and the fact that $u_x > 0$), at which point the chicken can exercise rule W_3 (it is easy to check that in the case where the x-component of the position of the car is greater than that of the chicken, F_3 is true; F_2 is always true since the chicken cannot move in the negative y-direction; finally, the chicken can simply choose its x-velocity v_x to be 0, which will satisfy $u_x \neq v_x$) until it reaches $y = H$. This shows that $p(0, 0, 0, c_x, c_y, 0)$ is true.

This proves that in this simple version of the "chicken-crossing-the-road" game, the chicken will always have a way to cross the road.

Note that a very easy modification of the program and proof above can allow us to place more restrictions on what the chicken can do and/or generalize on what the cars can do: for example, it is very easy to place a maximum and minimum on the components of the chicken's velocity with little extra work or to allow the cars to travel with any velocity (not just positive), which allows the model to become more realistic.

7.2 Many Cars, One Lane

Here, we prove the case where the Distributed Model with one lane in the road still always allows for the chicken to cross the road.

The previous version was very simple, and resultantly, the chicken had a very simple strategy. However, in the presence of many (possibly infinitely many) cars, the same strategy may not work. However, with the proper restrictions on the cars, we can guarantee that the chicken can make non-zero progress towards the other side of the road. Furthermore, the proof of the one car version is very instructive, in that we only need to be slightly modified its proof to generalize for the case of many cars.

The quantified hybrid program used here is the same as the one presented in the section on the Distributed Model in Hybrid Programs and

Safety Proof Details. We repeated the hybrid program here for convenience:

$$A \equiv k_x = 0 \wedge k_y = 0 \wedge k_z = 0 \wedge v_x = 0 \wedge v_y = 0 \wedge v_z = 0$$

$$B \equiv \forall i : C.\ 0 < c_y(i) - \frac{w}{2} \wedge c_y(i) + \frac{w}{2} < H \wedge c_z(i) = 0 \wedge u_x > 0$$

$$C \equiv \forall i : C.\ (\forall j : C.\ i \neq j \rightarrow (c_y(i) = c_y(j) \wedge (c_x(i) - c_x(j) < -l \vee c_x(i) - c_x(j) > l)))$$

$$D \equiv H > 0 \wedge w > 0 \wedge h \geq 0 \wedge l \geq 0 \wedge T > 0$$

$$\alpha \equiv k_z := h + 1; v_x := 0; v_y := 0; v_z := 0$$

$$\beta \equiv k_z := 0; v_x := *; v_y := *; v_z := 0; ?(v_y > 0); \gamma_1; \gamma_2; \mathsf{if}\ v_x = u_x\ \mathsf{then}\ ?(\forall i : C.\ k_x < c_x(i) - l \vee c_x(i) < k_x \vee t_1(i) > T)\ \mathsf{else}\ \gamma_3; \gamma_4; \delta_1; \delta_2\ \mathsf{fi}$$

$$\gamma_1 \equiv \forall i : C.\ t_1(i) := \frac{c_y(i) - \frac{w}{2} - k_y}{v_y}$$

$$\gamma_2 \equiv \forall i : C.\ t_2(i) := \frac{c_y(i) + \frac{w}{2} - k_y}{v_y}$$

$$\gamma_3 \equiv \forall i : C.\ s_1(i) := \frac{c_x(i) - k_x}{v_x - u_x}$$

$$\gamma_4 \equiv \forall i : C.\ s_2(i) := \frac{c_x(i) - l - k_x}{v_x - u_x}$$

$$\delta_1 \equiv ?\,(\forall i : C.\ s_1(i) \leq s_2(i) \rightarrow ((t_2(i) < s_1(i) \vee s_2(i) < t_1(i) \vee t_1(i) > T \vee s_1(i) > T))$$

$$\delta_2 \equiv ?\,(\forall i : C.\ s_1(i) > s_2(i) \rightarrow (t_2(i) < s_2(i) \vee s_1(i) < t_1(i) \vee t_1(i) > T \vee s_2(i) > T))$$

$$\epsilon \equiv \forall i : C.\ \{c_x(i)' = u_x, k_x' = v_x, k_y' = v_y, k_z' = v_z, t' = 1, t \leq T\}$$

$$\chi \equiv t := 0; (\alpha \cup \beta); \epsilon$$

Then we wish to prove the following:

$$A \wedge B \rightarrow \langle \chi^* \rangle\, k_y \geq H$$

We will prove this in multiple steps:

Let $p(\mathbf{k}, \{i \in C : \mathbf{c}(i)\})$ be the predicate $\langle \chi^* \rangle\, k_y \geq H$, where the vectors in the arguments have their obvious meanings. Here, the set $\{i \in C : \mathbf{c}(i)\}$ is taken to represent the vector/list of all the positions of all the cars, indexed by $i \in C$.

$$\langle * \rangle,\ \forall,\ MP\ \dfrac{US\ \dfrac{\langle ; \rangle,\ \langle := \rangle,\ \langle \cup \rangle\ \dfrac{\cdots\ \dfrac{Y_1}{\forall \mathbf{k}, \{i \in C : \mathbf{c}(i)\}\,(k_y \geq H \vee X_1 \vee X_2 \rightarrow p(\mathbf{k}, \{i \in C : \mathbf{c}(i)\})) \rightarrow (A \wedge B \wedge C \wedge D \rightarrow p(\mathbf{k}, \mathbf{c}))}}{\forall \mathbf{k}, \{i \in C : \mathbf{c}(i)\}\,((k_y \geq H \vee \langle \chi \rangle\, p(\mathbf{k}, \mathbf{c})) \rightarrow p(\mathbf{k}, \{i \in C : \mathbf{c}(i)\})) \rightarrow (A \wedge B \wedge C \wedge D \rightarrow p(\mathbf{k}, \{i \in C : \mathbf{c}(i)\}))}}{\forall \mathbf{k}, \{i \in C : \mathbf{c}(i)\}\,((k_y \geq H \vee \langle \chi \rangle\, \langle \chi^* \rangle\, k_y \geq H) \rightarrow \langle \chi^* \rangle\, k_y \geq H) \rightarrow (A \wedge B \wedge C \wedge D \rightarrow \langle \chi^* \rangle\, k_y \geq H)}}{A \wedge B \wedge C \wedge D \rightarrow \langle \chi^* \rangle\, k_y \geq H}$$

where

$$X_1 \equiv \{t := 0\}\,\langle\alpha\rangle\,\langle\epsilon\rangle\,p(\mathbf{k}, \{i \in C : \mathbf{c}(i)\})$$
$$X_2 \equiv \{t := 0\}\,\langle\beta\rangle\,\langle\epsilon\rangle\,p(\mathbf{k}, \{i \in C : \mathbf{c}(i)\})$$

Reduction of X_1:

Let

$$\zeta_1 \equiv \forall i : C.\ c_x(i) := c_x(i) + u_x \cdot \tilde{r}; k_x := k_x + v_x \cdot \tilde{r}; k_y := k_y + v_y \cdot \tilde{r}; k_z := k_z + v_z \cdot \tilde{r}; t := t + \tilde{r}$$
$$\zeta_2 \equiv \forall i : C.\ c_x(i) := c_x(i) + u_x \cdot r; k_x := k_x + v_x \cdot r; k_y := k_y + v_y \cdot r; k_z := k_z + v_z \cdot r; t := t + r$$

$$\langle;\rangle,\ \langle:=\rangle\ \frac{\langle'\rangle\ \dfrac{\mathit{subst}\ \dfrac{\mathbb{R}\ \dfrac{\exists 0 \le r \le T.\ p((k_x, k_y, h+1), \{i \in C : (c_x(i) + u_x \cdot r, c_y(i), c_z(i))\})}{\exists r \ge 0\,((\forall 0 \le \tilde{r} \le r\ \tilde{r} \le T) \wedge p((k_x, k_y, h+1), \{i \in C : (c_x(i) + u_x \cdot r, c_y(i), c_z(i))\}))}}{\{t := 0; k_z := h+1; v_x := 0; v_y := 0; v_z := 0\}\exists r \ge 0\,((\forall 0 \le \tilde{r} \le r\,\langle\zeta_1\rangle\,t \le T) \wedge \langle\zeta_2\rangle\,p(\mathbf{k}, \{i \in C : \mathbf{c}(i)\}))}}{\{t := 0; k_z := h+1; v_x := 0; v_y := 0; v_z := 0\}\,\langle\epsilon\rangle\,p(\mathbf{k}, \{i \in C : \mathbf{c}(i)\})}}{\{t := 0\}\,\langle\alpha\rangle\,\langle\epsilon\rangle\,p(\mathbf{k}, \{i \in C : \mathbf{c}(i)\})}$$

Reduction of X_2:

Let

$$\zeta_3 \equiv t := 0; k_z := 0; v_x := \widetilde{v_x}; v_y := \widetilde{v_y}; v_z := 0$$
$$\zeta_4 \equiv t := 0; k_z := 0; v_x := \widetilde{v_x}; v_y := \widetilde{v_y}; v_z := 0; \forall i : C.\ t_1(i) := \frac{c_y - \frac{w}{2} - k_y}{\widetilde{v_y}}; \forall i : C.\ t_2(i) := \frac{c_y + \frac{w}{2} - k_y}{\widetilde{v_y}}$$
$$\zeta_5 \equiv ?(\forall i : C.\ k_x < c_x(i) - l \vee c_x(i) < k_x \vee t_1(i) > T)$$

$$\langle;\rangle,\ \langle:=\rangle,\ \langle:*\rangle\ \frac{\langle?\rangle,\ \langle:=\rangle,\ \mathit{sub}\ \dfrac{\mathit{if},\ \mathit{sub},\ \exists\vee\ \dfrac{Z_1 \vee Z_2}{\exists \widetilde{v_x}, \widetilde{v_y}.\ \widetilde{v_y} > 0 \wedge \{\zeta_4\}\,\langle\mathsf{if}\ v_x = u_x\ \mathsf{then}\ \zeta_5\ \mathsf{else}\ \gamma_3; \gamma_4; \delta_1; \delta_2\ \mathsf{fi}\rangle\,\langle\epsilon\rangle\,p(\mathbf{k}, \{i \in C : \mathbf{c}(i)\}))}}{\exists \widetilde{v_x}, \widetilde{v_y}.\ \{\zeta_3\}\,\langle?(v_y > 0)\rangle\,\langle\gamma_1\rangle\,\langle\gamma_2\rangle\,\langle\mathsf{if}\ v_x = u_x\ \mathsf{then}\ \zeta_5\ \mathsf{else}\ \gamma_3; \gamma_4; \delta_1; \delta_2\ \mathsf{fi}\rangle\,\langle\epsilon\rangle\,p(\mathbf{k}, \{i \in C : \mathbf{c}(i)\}))}}{\{t := 0\}\,\langle\beta\rangle\,\langle\epsilon\rangle\,p(\mathbf{k}, \{i \in C : \mathbf{c}(i)\}))}$$

where $\widetilde{v_x}$ and $\widetilde{v_y}$ are fresh, and where

$$Z_1 \equiv \exists \widetilde{v_x}, \widetilde{v_y}.\ \widetilde{v_y} > 0 \wedge \widetilde{v_x} = u_x \wedge \{\zeta_4\}\,\langle\zeta_5\rangle\,\langle\epsilon\rangle\,p(\mathbf{k}, \{i \in C : \mathbf{c}(i)\})$$
$$Z_2 \equiv \exists \widetilde{v_x}, \widetilde{v_y}.\ \widetilde{v_y} > 0 \wedge \widetilde{v_x} \neq u_x \wedge \{\zeta_4\}\,\langle\gamma_3; \gamma_4; \delta_1; \delta_2\rangle\,\langle\epsilon\rangle\,p(\mathbf{k}, \{i \in C : \mathbf{c}(i)\})$$

Reduction of Z_1:

Let

$$F_1 \equiv \widetilde{v_y} > 0 \wedge \widetilde{v_x} = u_x \wedge \forall i : C.\ \left(k_x < c_x(i) - l \vee c_x(i) < k_x \vee \frac{c_y(i) - \frac{w}{2} - k_y}{\widetilde{v_y}} > T\right)$$

$$\dfrac{\dfrac{\dfrac{}{\exists \widetilde{v_x}, \widetilde{v_y}, 0 \le r \le T.\ (F_1 \wedge p((k_x + \widetilde{v_x} \cdot r, k_y + \widetilde{v_y} \cdot r, 0), \{i \in C : c_x(i) + u_x \cdot r, c_y(i), c_z(i)\}))}\ ax}{\exists \widetilde{v_x}, \widetilde{v_y}.\ (F_1 \wedge \exists 0 \le r \le T.\ p((k_x + \widetilde{v_x} \cdot r, k_y + \widetilde{v_y} \cdot r, 0), \{i \in C : c_x(i) + u_x \cdot r, c_y(i), c_z(i)\}))}\ \langle' \rangle,\ sub,\ \mathbb{R}}{\exists \widetilde{v_x}, \widetilde{v_y}.\ \widetilde{v_y} > 0 \wedge \widetilde{v_x} = u_x \wedge \forall i : C.\ \left(k_x < c_x(i) - l \vee c_x(i) < k_x \vee \frac{c_y(i) - \frac{w}{2} - k_y}{\widetilde{v_y}} > T\right) \wedge \{\zeta_4\}\, \langle \epsilon \rangle\, p(\mathbf{k}, \{i \in C : \mathbf{c}(i)\})}\ \langle ? \rangle, \langle := \rangle,\ sub$$

$$\overline{\exists \widetilde{v_x}, \widetilde{v_y}.\ \widetilde{v_y} > 0 \wedge \widetilde{v_x} = u_x \wedge \{\zeta_4\}\, \langle \zeta_5 \rangle\, \langle \epsilon \rangle\, p(\mathbf{k}, \{i \in C : \mathbf{c}(i)\})}$$

Reduction of Z_2:

Let

$$\zeta_6 \equiv t := 0; k_z := 0; v_x := \widetilde{v_x}; v_y := \widetilde{v_y}; v_z := 0; t_1(i) := \frac{c_y(i) - \frac{w}{2} - k_y}{\widetilde{v_y}}; t_2(i) := \frac{c_y(i) + \frac{w}{2} - k_y}{\widetilde{v_y}}; s_1(i) := \frac{c_x(i) - k_x}{\widetilde{v_x} - u_x}; s_2(i) := \frac{c_x(i) - l - k_x}{\widetilde{v_x} - u_x}$$

$$\widetilde{t_1}(i) \equiv \frac{c_y(i) - \frac{w}{2} - k_y}{\widetilde{v_y}}$$

$$\widetilde{t_2}(i) \equiv \frac{c_y(i) + \frac{w}{2} - k_y}{\widetilde{v_y}}$$

$$\widetilde{s_1}(i) \equiv \frac{c_x(i) - k_x}{\widetilde{v_x} - u_x}$$

$$\widetilde{s_2}(i) \equiv \frac{c_x(i) - l - k_x}{\widetilde{v_x} - u_x}$$

$$\dfrac{\dfrac{\dfrac{}{Z_3}}{\exists \widetilde{v_x}, \widetilde{v_y}.\ \widetilde{v_y} > 0 \wedge \widetilde{v_x} \neq u_x \wedge G_1 \wedge G_2 \wedge \{\zeta_6\}\, \langle \epsilon \rangle\, p(\mathbf{k}, \{i \in C : \mathbf{c}(i)\})}\ \langle' \rangle,\ sub,\ \mathbb{R},\ \exists \wedge}{\exists \widetilde{v_x}, \widetilde{v_y}.\ \widetilde{v_y} > 0 \wedge \widetilde{v_x} \neq u_x \wedge \{\zeta_6\}\, \langle \delta_1 \rangle\, \langle \delta_2 \rangle\, \langle \epsilon \rangle\, p(\mathbf{k}, \{i \in C : \mathbf{c}(i)\})}\ \langle ? \rangle, \rightarrow,\ sub$$

$$\overline{\exists \widetilde{v_x}, \widetilde{v_y}.\ \widetilde{v_y} > 0 \wedge \widetilde{v_x} \neq u_x \wedge \{\zeta_4\}\, \langle \gamma_3; \gamma_4; \delta_1; \delta_2 \rangle\, \langle \epsilon \rangle\, p(\mathbf{k}, \{i \in C : \mathbf{c}(i)\})}\ \langle ; \rangle, \langle := \rangle,\ sub$$

where

$$
\begin{aligned}
Z_3 &\equiv \exists \widetilde{v_x}, \widetilde{v_y}, 0 \leq r \leq T.\ \widetilde{v_y} > 0 \wedge \widetilde{v_x} \neq u_x \wedge G_1 \wedge G_2 \wedge p((k_x + \widetilde{v_x} \cdot r, k_y + \widetilde{v_y} \cdot r, 0), \{i \in C : (c_x(i) + u_x \cdot r, c_y, c_z)\}) \\
G_1 &\equiv \forall i : C.\ \widetilde{s_1}(i) \leq \widetilde{s_2}(i) \rightarrow \left(\widetilde{t_2}(i) < \widetilde{s_1}(i) \vee \widetilde{s_2}(i) < \widetilde{t_1}(i) \vee \widetilde{t_1}(i) > T \vee \widetilde{s_1}(i) > T\right) \\
G_2 &\equiv \forall i : C.\ \widetilde{s_1}(i) > \widetilde{s_2}(i) \rightarrow \left(\widetilde{t_2}(i) < \widetilde{s_2}(i) \vee \widetilde{s_1}(i) < \widetilde{t_1}(i) \vee \widetilde{t_1}(i) > T \vee \widetilde{s_2}(i) > T\right)
\end{aligned}
$$

Proof of Y_1:

Let

$$
\begin{aligned}
W_1 &\equiv \exists 0 \leq r \leq T.\ p((k_x, k_y, h+1), \{i \in C : (c_x(i) + u_x \cdot r, c_y(i), c_z(i))\}) \\
W_2 &\equiv \exists \widetilde{v_x}, \widetilde{v_y}, 0 \leq r \leq T.\ \ (F_1 \wedge p((k_x + \widetilde{v_x} \cdot r, k_y + \widetilde{v_y} \cdot r, 0), \{i \in C : c_x(i) + u_x \cdot r, c_y(i), c_z(i))})) \\
W_3 &\equiv \exists \widetilde{v_x}, \widetilde{v_y}, 0 \leq r \leq T.\ \widetilde{v_y} > 0 \wedge \widetilde{v_x} \neq u_x \wedge G_1 \wedge G_2 \wedge p((k_x + \widetilde{v_x} \cdot r, k_y + \widetilde{v_y} \cdot r, 0), \{i \in C : (c_x(i) + u_x \cdot r, c_y, c_z)\}) \\
E &\equiv A \wedge B \wedge C \wedge D
\end{aligned}
$$

Now, we can continue on with the first proof:

$$
\frac{\rightarrow \dfrac{\forall \mathbf{k}, \{i \in C : \mathbf{c}(i)\}\,((k_y \geq H \vee W_1 \vee W_2 \vee W_3) \rightarrow p(\mathbf{k}, \{i \in C : \mathbf{c}(i)\})) \rightarrow (\neg(E) \vee (E \wedge p(\mathbf{k}, \{i \in C : \mathbf{c}(i)\})))}{\forall \mathbf{k}, \{i \in C : \mathbf{c}(i)\}\,((k_y \geq H \vee W_1 \vee W_2 \vee W_3) \rightarrow p(\mathbf{k}, \{i \in C : \mathbf{c}(i)\})) \rightarrow (E \rightarrow p(\mathbf{k}, \{i \in C : \mathbf{c}(i)\}))}}{\forall \mathbf{k}, \{i \in C : \mathbf{c}(i)\}\,((k_y \geq H \vee \{t := 0\}\,\langle\alpha\rangle\,\langle\epsilon\rangle\,p(\mathbf{k}, \{i \in C : \mathbf{c}(i)\}) \vee \{t := 0\}\,\langle\beta\rangle\,\langle\epsilon\rangle\,p(\mathbf{k}, \{i \in C : \mathbf{c}(i)\})) \rightarrow p(\mathbf{k}, \{i \in C : \mathbf{c}(i)\})) \rightarrow (E \rightarrow p(\mathbf{k}, \{i \in C : \mathbf{c}(i)\})}
$$

As a reminder, we have

$$
\begin{aligned}
A &\equiv k_x = 0 \wedge k_y = 0 \wedge k_z = 0 \wedge v_x = 0 \wedge v_y = 0 \wedge v_z = 0 \\
B &\equiv \forall i : C.\ 0 < c_y(i) - \frac{w}{2} \wedge c_y(i) + \frac{w}{2} < H \wedge c_z(i) = 0 \wedge u_x > 0 \\
C &\equiv \forall i : C.\ (\forall j : C.\ i \neq j \rightarrow (c_y(i) = c_y(j) \wedge (c_x(i) - c_x(j) < -l \vee c_x(i) - c_x(j) > l))) \\
D &\equiv H > 0 \wedge w > 0 \wedge h \geq 0 \wedge l \geq 0 \wedge T > 0 \\
F_1 &\equiv \widetilde{v_y} > 0 \wedge \widetilde{v_x} = u_x \wedge \forall i : C.\ \left(k_x < c_x(i) - l \vee c_x(i) < k_x \vee \frac{c_y(i) - \frac{w}{2} - k_y}{\widetilde{v_y}} > T\right)
\end{aligned}
$$

If the left hand side $\forall \mathbf{k}, \{i \in C : \mathbf{c}(i)\}\,((k_y \geq H \vee W_1 \vee W_2 \vee W_3 \vee W_4) \rightarrow p(\mathbf{k}, \{i \in C : \mathbf{c}(i)\}))$ implies $\neg E$, then there is nothing to show, so we may instead simply prove

$$
E \wedge \forall \mathbf{k}, \{i \in C : \mathbf{c}(i)\}\,((k_y \geq H \vee W_1 \vee W_2 \vee W_3 \vee W_4) \rightarrow p(\mathbf{k}, \{i \in C : \mathbf{c}(i)\})) \rightarrow p(\mathbf{k}, \{i \in C : \mathbf{c}(i)\})
$$

Then the proof continues as:

$$eq\ \frac{E \wedge \forall \mathbf{k}, \{i \in C : \mathbf{c}(i)\}\ ((k_y \geq H \vee W_1 \vee W_2 \vee W_3 \vee W_4) \rightarrow p(\mathbf{k}, \{i \in C : \mathbf{c}(i)\})) \rightarrow p((0,0,0), \{i \in C : c_x(i), c_y(i), 0)\})}{E \forall \mathbf{k}, \{i \in C : \mathbf{c}(i)\}\ ((k_y \geq H \vee W_1 \vee W_2 \vee W_3 \vee W_4) \rightarrow p(\mathbf{k}, \{i \in C : \mathbf{c}(i)\})) \rightarrow p(\mathbf{k}, \{i \in C : \mathbf{c}(i)\})}$$

Now, we can consider the W_i's as "rules" that decide which values of $\mathbf{k}$ and $\mathbf{c}$ makes $p(\mathbf{k}, \mathbf{c})$ true, and prove efficiency by working backwards.

Rule W_1 allows the chicken to simply hover in place at a greater height; from the rule, one should note that this move does not actually help the chicken directly make progress towards the goal.

Rule W_2 allows the chicken to move at a constant non-zero velocity that it chose (such that its x-velocity matches the velocity of the cars, and such that it will not collide with any one of the cars within a period of T time; this is determined by calculating the time $t_1(i)$ that it would take for the chicken to enter the path of car i and checking that time is greater than T, and by making sure that the chicken is not travelling along side of a car, in which case the chicken is guaranteed to hit the car given enough time).

Rule W_3 allows the chicken to move at a constant non-zero velocity that it chose (such that its x-velocity is different from the velocity of the cars, and such that it will not collide with any one of the cars within a period of T time; this is done by calculating the times $t_1(i)$ and $t_2(i)$ it takes the chicken to enter and exit the path of car i and the times $s_1(i)$ and $s_2(i)$ it takes for the car i's length to coincide with the chicken's x position, and determining if the interval with endpoints $t_1(i)$ and $t_2(i)$ intersects the interval with endpoints $s_1(i)$ and $s_2(i)$.

Initially, the chicken starts at $(0, 0, 0)$ and each car i at $(c_x(i), c_y(i), 0)$. From C, we know that the $c_y(i)$'s are equal; let us denote their common value by c_y, so that each car starts at $(c_x(i), c_y, 0)$.

Now, one strategy the chicken can follow is this: wait (hover) until there is no car on the line $x = 0$, and then zoom across at a high velocity. This is a safe and sure way of reaching the goal (easy applications of rule W_1 and W_3).

Of course, this is again not very realistic. However, with this proof in hand, one can now easily add restrictions and modify the constraints of the chicken's velocity, without disrupting the existence of a solution for the chicken to cross the road.

One can easily add a cap on the chicken's velocity. Then it is not hard to see that if the chicken ever ends up in the path of a car, but cannot cross that path quickly enough, it can choose to jump and hover until the car has passed, and then continue on its merry way. Since C guarantees that every car is separated from each other by at least some gap, it is always possible for the chicken to do this.

Again, from what we have so far, it is also not difficult to see that the same problem, except allowing for varying lengths and widths and heights of the cars (so car i has width $w(i)$, length $l(i)$, and height $h(i)$), still has a solution for the chicken, *as long as the chicken can choose the height at which it will hover.*

7.3 Many Cars, Many Lanes

The next step from the previous scenario is to include many lanes where cars may drive. Lanes will have a common width ω, and every car is guaranteed to fit inside a line, i.e., $w(i) < \omega$. Here, cars in different lanes will be guaranteed to be separated width-wise, i.e. if car i and j are

in different lanes, then we guarantee that $|c_y(i) - c_y(j)| > \frac{w(i)+w(j)}{2}$, and that $c_y(i)$ is equal to some fixed value(s) of y that is predetermined beforehand (for example, we can have three different values to model a road with three lanes). Because the proof for this situation is so similar to the proof for the case of **Infinitely Long Road, One Lane, Many Cars**, we omit it.

7.4 Many Cars, Many Lanes, No Jump

Finally, a modification to the choices the chicken can make can be made so that the chicken can no longer instantaneously jump, which is more true to physics.

If we allow a sufficient gap between the cars, and have a cap on the velocities of the cars (i.e., a road speed limit), it is easy to see that the chicken will still have a way to safely reach the goal on the other side of the road. Formally, if we place a speed limit S on the cars, and the maximum velocity of the chicken is M, then by having

$$|c_x(i) - l - c_y(i)| \geq S \cdot \frac{\omega}{M},$$

then the chicken has will be able to cross a lane safely (this equation represents the fact that the time it takes for a car to travel the gap between cars is at least the time it takes the chicken to cross the width of the cars w). Furthermore, since there are gaps between cars in different lanes, the chicken can choose to squat in between cars in the between lanes (i.e., choose velocity 0).

Again, the formal proof of the program for this case is similar to the proof of **Infinitely Long Road, One Lane, Many Cars**, we omit it.