Recitation 5: Time-Triggered Control and Differential Invariants
15-424/15-624/15-824 Logical Foundations of Cyber-Physical Systems
Notes by Yong Kiam Tan, edits by Katherine Cordwell (kcordwel@andrew.cmu.edu)

1 Announcements

- Theory 3 is now available.
- Theory 2 will be graded soon, watch on Piazza for announcements when feedback is available.

2 Review: Lab 1 Veribot

In recitation we discussed common mistakes on the assignments, but we will not give out solutions in the recitation notes. If you have questions, come talk to us.

3 Motivation and Learning Objectives

For the first half of this recitation, we will work on changing the event-triggered model from the last recitation into a time-triggered one. Recall that at the end of last recitation, we considered refining the event-triggered model to be more realistic by making the players unable to react when the velocity of the ball is too high. Time-triggered control is another refinement of the event-triggered model: it models the periodic nature of the controller (in this case, the players).

Both the event- and time-triggered controllers were proved by solving the differential equations. For more complicated ODEs we will not, in general, be able to solve them directly. The second half of the recitation starts looking at *differential invariants*, which is a proof technique for proving properties about differential equations without solving them.

4 Time-Triggered Ping Pong

As a reminder of the model from last week, the players Forrest and Dan are at positions l, r respectively, the ball's position is x and its velocity is v. The goal is to show the safety postcondition $l \leq x \leq r$, i.e., the ball stays between the two players no matter how many times the players hit the ball.

To simplify matters, we shall forget about the players Forrest and Dan, and instead think of both players as controlled by a central computer which is hitting the ball. This computer has a control loop that runs every $T > 0$ seconds, which allows it to fire the ping pong paddles to hit the ball back from either the left or right boundaries. One possible shape for a model of such a controller is:

$$(Ctrl; Plant)^*$$

where $Ctrl$ models some discrete decisions made by the controller, $Plant$ models the continuous physics that happens when the controller is sleeping, and the loop models the controller's periodic behavior.

Our plant model for continuous physics is relatively simple. We will just need to add a timer to the motion equation for the ball:

$$Plant \overset{\text{def}}{\equiv} t := 0; \{x' = v, t' = 1 \,\&\, t \leq T\}$$

Contrast this with last week's event-triggered model, where we added a choice between two differential equations to account for the event when $x = l \vee x = r$. Here, there is only one system of differential equations, where the continuous physics is allowed to run for any duration $0 \leq t \leq T$. This non-determinism means that the time bound T should be more accurately thought of as an *upper bound* on how long the controller can sleep. This makes physical sense: we can never be exactly precise about how long a controller will sleep for, but we will certainly often be able to give a limit on the duration. Like last week, our goal is to prove a formula of the form:

$$Pre \rightarrow [(Ctrl; Plant)^*] l \leq x \leq r$$

where $Pre, Ctrl$ are the parts of the time-triggered model that we will determine next.

4.1 Time-Triggered Control (Single Control Cycle)

Thinking about worst-case scenarios is useful when designing time-triggered controllers. Here, one way of figuring out the ping pong controller is to put ourselves in its shoes and ask the following question:

"If I waited another control cycle of T seconds, would it be too late to react?"

If we let $Plant$ execute, then (in the worst case) the controller might only be able to react T seconds later. The controller must therefore ensure that whatever control decision it makes would allow it to react safely in the next cycle even in this worst case. To understand the potential issues, let us first remove the loop and focus on one control cycle:

$$Pre \rightarrow [Ctrl; Plant] l \leq x \leq r$$

To get us started, here is the controller from last week:

$$Ctrl_{event} \overset{\text{def}}{\equiv} \textbf{if } x = l \wedge v \leq 0 \vee x = r \wedge v \geq 0 \textbf{ then } v := -v$$

Note: Remember that "if P then α" can be expressed in hybrid programs as $?P; \alpha \cup ?\neg P$. The $?\neg P$ is very important for avoiding an overly constrained controller/vacuous behavior. I didn't do a good job of addressing this in class, so please ensure that you see why.

Exercise 1:
Is this controller safe for our time-triggered model?

Answer: It is clearly not going to be safe. For example, if x was already very close but not equal to r, and it was flying with velocity $v \geq 0$ then this controller we will not react. In that case, if $x + vT > r$ then the ball could have violated the right bound before we next get a chance to react.

Similarly, if the ball was flying with velocity $v \leq 0$, and $x + vT < l$, then the ball could leave the left bound. Therefore, our first fix to the controller shall be to make sure that we react in either of the scenarios described above:

$$Ctrl \stackrel{\text{def}}{\equiv} \textbf{if } x + vT < l \wedge v \leq 0 \vee x + vT > r \wedge v \geq 0 \textbf{ then } v := -v$$

Notice that adding time-triggered control has forced us to make our physical interpretation of the model more realistic. We previously said that the players were exactly standing at positions $x = l$ and $x = r$ and could only hit the ball at those positions. As we just saw, however, it is impossible to make such a system safe with time-triggered control – we must react earlier than when the ball is already at the boundaries.

Now that we have a candidate controller, let us start with a simple precondition where the ball is initially between the two boundaries and flying towards the right boundary, i.e.,

$$Pre \stackrel{\text{def}}{\equiv} l \leq x \leq r \wedge v \geq 0 \wedge T > 0$$

Exercise 2:
Is the system safe?
Answer: Not quite. The problem is that the ball could already be traveling so fast that, for example, if we hit it back near the right boundary, it will leave the left boundary before we can react!

This is not a problem that can be solved directly by changing the controls.[1] We already know from our earlier argument that the controller *must* react and hit the ball when it is too close to the right boundary. Thus, we will need additional assumptions about the system.

Exercise 3:
What assumptions should we add to the system?
Answer: Suppose that the ball is heading towards the right boundary with $v \geq 0$. By the earlier argument, we may be hitting the ball as early as when it is at position $r - vT$. Next, the ball will fly with negative velocity towards the left boundary but we the controller may not get to respond until T seconds later. Thus, the ball might reach $r - 2vT$ and so in order for the system to be safe, we need to assume that the boundaries are at least separated by $l \leq r - 2vT$.

4.2 Time-Triggered Control (Loop)

Given the above controller and assumptions for a single control cycle, it is time to see if our reasoning also works when we add back the loop. To be precise, here is the formula with all

[1] Unless we also allow the controller to set velocity explicitly rather than just flipping its direction.

the abbreviations completely expanded out:

$$
\begin{array}{l}
l \leq x \leq r \wedge v \geq 0 \wedge T > 0 \wedge l + 2vT \leq r \\
\big[\big(\textbf{if } x + vT < l \wedge v \leq 0 \vee x + vT > r \wedge v \geq 0 \textbf{ then } v := -v; \\
t := 0; \{x' = v, t' = 1 \,\&\, 0 \leq t \leq T\}\big)^*\big] \\
l \leq x \leq r
\end{array}
$$

Exercise 4:
Is the controller (and preconditions) safe for the time-triggered model (with loops)? If so, what loop invariant should we use in its proof?
Answer: Yes, the loop invariant is $l \leq x \leq r \wedge l + 2|v|T \leq r$. This model is rather special because we only needed to look ahead for one control cycle. In general, you may need to consider the effect that delaying your reaction for one control cycle could have on safety for the *entire loop*.
Note: In class, we noted that the time-triggered car model in Lab 2 is indeed an example where you need to consider the effect of your control decision for all future loop iterations. Intuitively, this is because controlling acceleration has an "inertial" effect on velocity.
Note: In class, we briefly looked at a model of the new time-triggered controller in KeYmaera X. The model is called rec5time in the archive. We went through the proof very quickly, so maybe you will want to go through it more slowly on your own.

The proof of this time-triggered controller (and last week's event-triggered controller) both make use of KeYmaera X's built-in ODE solving capabilities. Unfortunately, not many ODEs are solvable in closed form with polynomial solutions (if they are solvable at all). The ones that do, like in our current ping-pong model, are rather boring from a modeling perspective. Further, although KeYmaera X's ability to solve ODEs has improved significantly from previous versions, dL provides much more efficient ODE reasoning techniques for ODEs than directly solving them.

As a sneak preview, the archive has a proof of the same model but only replacing the tactic that solves ODEs with differential invariants. Try out both proofs on your machine and see if there is a difference in speed. On the TA's machine, they are both quite fast (but again, the differential equations are quite simple here, so solving them doesn't introduce a huge burden).

We could also imagine refining our model of physics to account for air resistance when the ball is in flight, for example by adding a drag on the velocity:

$$x' = v, v' = -kv^2$$

Exercise 5:
If you are feeling brave, solve the above ODEs explicitly for the closed form general solution. To simplify matters, take $k = 1$ and the initial velocity $v_0 \geq 0$.

Answer:

Solving for velocity first yields:

$$v(t) = \frac{v_0}{1 + v_0 t}$$

Solving for position yields:

$$x(t) = \ln(v_0 t + 1) + x_0$$

Notice that the solutions are much complicated than what we started with: they are not even expressible using polynomials. If you thought solving these equations was complicated, think about how you could formally convince a computer that these are the solutions, and that these solutions imply your safety property!

In fact, if $v_0 \leq 0$, then the solution above does not even exist for all time. Fortunately, for us, that situation is not physically possible because air resistance acts in the opposite direction to velocity. However, it is yet another reason why simply attempting to solve the ODEs symbolically for a global solution will not work.

We will not (yet) be able to prove properties for the more advanced model of physics but we will start getting there by looking at *differential invariants*.

5 Differential Invariants

The central insight behind the ODE reasoning principles in dL is that we can work *directly with the differential equations* rather than their solutions. In order to do this formally, we will need a way to work syntactically with derivatives.

5.1 Syntax and Semantics

Let us start by extending the syntax of dL terms with a case for differential variables x' and one for differential terms $(e)'$:

$$e ::= \cdots \mid x' \mid (e)'$$

Of course, once we add syntax, we also need semantics—that is, we need some meaning for $\omega[\![x']\!]$ and $\omega[\![(e)']\!]$, where ω is an arbitrary state. But this isn't actually so easy. We want differential terms and variables to have a meaning that corresponds to the reason we added them in the first place—so, for example, intuitively we want $(e)'$ to take the value of the derivative of term e along the solution to a differential equation.

A question immediately arises: which ODE are we talking about? We are defining the semantics of terms $\omega[\![e]\!]$ with respect to some given state ω without reference to any specific ODE. Should we re-define our semantics to account for ODEs? Well, then which ODE should we use for the ordinary terms that do not mention any derivatives? The solution to all of these questions is to move to a differential-form understanding of the differential terms.

The key insight is to insist that states give value to the differential variables x'. (That is, we can look up the value of x' in ω in the same way that we can look up the value of variable x in state ω.) Then we'll define the semantics of the $\omega[\![(e)']\!]$ using the $\omega[\![x']\!]$ as building blocks.

Moreover, whenever we have an ODE in mind, we want the states along the solution to capture the meaning of the relevant differential variables in a meaningful way—i.e. a way that is directly given by the ODE. This is rigorized in the semantcs of ODEs.

$$[[x' = f(x)\ \&\ Q]] = \{(\omega,\nu) : \phi(0) = \omega \text{ except at } x' \text{ and } \phi(r) = \nu \text{ for a solution } \phi : [0,r] \to \mathcal{S} \text{ of duration } r \text{ satisfying } \phi \models x'{=}f(x) \wedge Q, \}$$

where $\phi \models x'{=}f(x) \wedge Q$ means for all times $0{\leq}z{\leq}r$, $\phi(z) \in [[x'{=}f(x) \wedge Q]]$ with $\phi(z)(x') = \frac{d\phi(t)(x)}{dt}(z)$ and $\phi(z) = \phi(0)$ except at x, x'

Let's break this definition down a bit and see what it tells us about the meaning of differential variables (with some color-coding!):

- As before, variables that aren't changing in the ODE are constant
- The initial value of x' in ω doesn't matter because we don't really know anything about what x' means in an arbitrary ω. It's only once we're along the solution ϕ that x' has a meaning connected to the ODE—so in $\phi(0)$, x' suddenly takes a meaning connected to the ODEs.
- The initial value of x does matter—that's our initial condition. So $\phi(0)$ and ω must agree on the value of x.
- x' has the meaning we want along the ODE—this is captured in $\phi \models x'{=}f(x) \wedge Q$.
- The value of x is updated per the value of x' along ϕ
- Oh, and we have a domain constraint Q which behaves as usual.

To recap: we're treating $\omega[\![x']\!] = \omega(x')$ as a variable lookup, and saying that these x' take on special meaning in states along a solution to an ODE (per the above definition of the semantics of ODEs). In an arbitrary state, we can't really ascribe any significance to the value of $\omega(x')$. And now that we really know what the differential variables mean, we're ready to turn to differential terms. The semantics of a differential term is defined as follows:

$$\omega[\![(e)']\!] = \sum_{x \in \mathbb{V}} \omega(x') \cdot \frac{\partial [\![e]\!]}{\partial x}(\omega)$$

Let us break down the various parts of this definition step-by-step.

1. Differential variable lookup $\omega(x')$.

   To formally understand this, we will now require that states ω not only map variables x to real values, but also all of their corresponding primed symbols variables x' to real values as well. Thus, $\omega(x')$ simply means look up the value of variable x' in state ω.

2. Partial derivative $\frac{\partial [\![e]\!]}{\partial x}$.

   Recall that $[\![e]\!]$ is a function $\mathcal{S} \to \mathbb{R}$ it therefore makes sense for us to consider its partial (spatial) derivatives with respect to the input coordinates.[2] It is just like any other multivariate function, e.g., $f : \mathbb{R}^2 \to \mathbb{R} \overset{\text{def}}{\equiv} x^2 + y^2$, with $\frac{\partial f}{\partial x} = 2x$, $\frac{\partial f}{\partial y} = 2y$, where $\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y}$ are both functions $\mathbb{R}^2 \to \mathbb{R}$.

   All of the partial derivatives, $\frac{\partial [\![e]\!]}{\partial x}$, are again functions $\mathcal{S} \to \mathbb{R}$.

3. Evaluating partial derivative at ω, i.e., $\frac{\partial [\![e]\!]}{\partial x}(\omega)$.

   As we have just seen, $\frac{\partial [\![e]\!]}{\partial x} : \mathcal{S} \to \mathbb{R}$ maps an input state to a real value.

4. $\omega(x') \cdot \frac{\partial [\![e]\!]}{\partial x}(\omega)$

   Given that both $\omega(x')$ and $\frac{\partial [\![e]\!]}{\partial x}(\omega)$ are real values, it now makes sense for us to talk about their product, which will be another real number. We will better understand the use of this product shortly.

5. $\sum_{x \in \mathbb{V}} \omega(x') \cdot \frac{\partial [\![e]\!]}{\partial x}(\omega)$

   Finally, all of the above was done for a single variable $x \in \mathbb{V}$. The semantics of $\omega[\![(e)']\!]$ is to sums over all possible variables. This is well-defined because there are only finitely many variables that can be mentioned in any term e. The spatial derivative $\frac{\partial [\![e]\!]}{\partial z}$ for variables z not mentioned in e is simply 0 and can be dropped from the sum.

5.2 The Differential Lemma and the Differential Assignment Lemma

Solutions of differential equations are where our new understanding of differential terms will be put to use. Consider a solution $\varphi : [0, T] \to \mathcal{S}$ with $T > 0$ and $\varphi \models \{x' = f(x)\} \wedge Q$. Since each time $\varphi(t)$ is a state, it makes sense to consider the value of the term e along the solution as a function of time: $\varphi(t)[\![e]\!]$. Now, let us consider the *time derivative* of this function, which tells us how the value of e changes along the solution:

$$\frac{d\varphi(t)[\![e]\!]}{dt}$$

By applying the multivariate the chain rule, for time τ:

$$\frac{d\varphi(t)[\![e]\!]}{dt}(\tau) = \sum_{x \in \mathbb{V}} \frac{\partial [\![e]\!]}{\partial x}(\varphi(\tau)) \cdot \frac{d\varphi(t)(x)}{dt}(\tau)$$

But by definition of a solution, we already know $\frac{d\varphi(t)(x)}{dt}(\tau) = \varphi(\tau)(x')$, and so:

$$\frac{d\varphi(t)[\![e]\!]}{dt}(\tau) = \sum_{x \in \mathbb{V}} \frac{\partial [\![e]\!]}{\partial x}(\varphi(\tau)) \cdot \varphi(\tau)(x')$$

[2] Provided that the $[\![e]\!]$ is sufficiently smooth so that all of its partial derivatives exist.

Finally, by definition, the RHS is the differential term $(e)'$ evaluated at $\varphi(\tau)$:

$$\frac{d\varphi(t)[\![e]\!]}{dt}(\tau) = \varphi(\tau)[\![(e)']\!]$$

The last equality yields the crucial *differential lemma*, i.e., that the value of the differential e' coincides with the time-derivative of the value of e along solutions of an ODE.

This lemma gives us license to work directly with differentials when we want to reason about the derivatives of terms along solutions to an ODE. Differentials are useful because they have a well-defined semantics that is independent of the ODEs.

Note: In class we didn't talk about the derivation of the differential lemma.

In addition to the differential lemma, we have the differential assignment lemma: If $\phi \models x'=f(x) \land Q$ then $\phi \models P \leftrightarrow [x' := f(x)]P$.

Question: Why does this make sense?

Answer: Intuitively, we've just seen how along solutions, differentials mean what we expect them to. Slightly more rigorously, this again comes from the semantics of ODEs.

Question: Why is this useful?

Answer: Very informally, we may find it helpful to remove the funny primes because that helps us safely elide the ODEs.

5.3 Examples

Note: This section gives more details than we had time for in recitation.

Let's do some concrete examples! Say we have the ODE $x' = 1$ and initially $x = 0$.

Say that we want to evaluate x' in the state $\phi(k)$, where ϕ is the (unique, global) solution to $x' = 1$. Well, we know that $\phi(k)[[x']] = \phi(k)(x')$, because as discussed before, differential variables are treated like variables (i.e. their evaluation in a state is just variable lookup). Now, using the semantics of differential equations, $\phi(k) \in [[x' = 1]]$, so $\phi(k)(x') = 1$.

How do we calculate what $\phi(k)(x)$ is? We know from the semantics of differential equations that $\frac{d\phi(t)(x)}{dt}(k) = 1$. Because k was arbitrary, this tells us that $\frac{d\phi(t)(x)}{dt} = 1$, or $\phi(t)(x) = t$, so $\phi(k)(x) = k$. Another way of calculating $\phi(k)(x)$ would be to notice that the solution to the ODE is $\phi(t) = \{x \mapsto t\}$, so $\phi(0)(x) = 0$, $\phi(1)(x) = 1$, $\phi(t)(x) = t$, etc—and so in particular, $\phi(k)(x) = k$. Notice that these calculations give the same value (as they should—it's a good sanity check).

Now let's say that we want to evaluate the term $(x^2+2x)'$ in the state $\phi(k)$. One way is to use the differential lemma: $\phi(k)[[(x^2+2x)']] = \frac{d\phi(t)[[x^2+2x]]}{dt}(k)$. Now, $\phi(t)[[x^2+2x]] = \phi(t)(x) \cdot \phi(t)(x)+2\phi(t)(x)$, and we've just calculated that $\phi(t)(x) = t$. So $\phi(t)[[x^2+2x]] = t^2+2t$, and now plugging this into the differential lemma, $\frac{d\phi(t)[[x^2+2x]]}{dt}(k) = \frac{d(t^2+2t)}{dt}(k) = (2t+2)(k) = 2k+2$.

Or, if we didn't use the differential lemma, we could instead apply the definition directly and calculate $\phi(k)[[(x^2+2x)']] = \phi(k)(x') \cdot \frac{\partial[[x^2+2x]]}{\partial x}(\phi(k))$. Here, to be fully precise, $[[x^2+2x]]$

is a function from the set of states to $\mathbb{R}$, so its derivative is also a function from the set of states to $\mathbb{R}$ which we are evaluating at the particular state $\phi(k)$. In $\phi(k)$, x evaluates to k, so when we evaluate the function $\omega \mapsto 2\omega(x) + 2$ at $\phi(k)$ we get $2k + 2$—which aligns with our previous calculations.

5.4 Differential Axioms

Lemmas are great, but what's even more useful? AXIOMS, of course! The differential axioms of dL allow us to manipulate differential terms. Given the differential lemma, the shape of these axioms should be unsurprising, because they remind us precisely of the standard rules for working with derivatives:

$$
\begin{aligned}
(c())' &= 0 \\
(x)' &= x' \\
(f+g)' &= (f)' + (g)' \\
(f \cdot g)' &= (f)' \cdot g + f \cdot (g)' \\
\left(\frac{f}{g}\right)' &= \frac{(f)'g - f(g)'}{g^2}
\end{aligned}
$$

Using these axioms in the concrete example we've just discussed in Section 5.3 gives: $\phi(k)[[2x \cdot x' + 2x']] = 2\phi(k)[[x]] \cdot \phi(k)[[x']] + 2\phi(k)[[x']]$. Remember that we know that $\phi(k)[[x']] = \phi(k)(x')$ and we previously calculated that $\phi(k)(x') = 1$ and $\phi(k)(x) = t$. Plugging these in gives $\phi(k)[[(x^2+2x)']] = 2k+2$, which agrees with our previous calculations.

In addition, the following *differential effect* axiom is sound:

$$\text{(DE)} \quad [\{x' = f(x)\,\&\,Q\}]P \leftrightarrow [\{x' = f(x)\,\&\,Q\}][x' := f(x)]P$$

Axiom DE allows us to replace the postcondition P on the left with a box differential assignment $[x' := f(x)]P$. It is sound because we required that the differential variables x' take on the values of the RHS along solutions of an ODE. Intuitively, DE is much like the differential assignment lemma, but as an axiom.

We also have the DI axiom:

$$\text{(DI)} \quad ([x' = f(x)]e = 0 \leftrightarrow e = 0) \leftarrow [x' = f(x)](e)' = 0$$

Intuitively this is saying that if the rate of change of e along the differential equation is 0, then e stays constant. You could actually replace the 0's on the LHS with any constant c. **Note: We will cover the DE axiom (and the DI axiom) in more detail next week after the lectures on the rest of the ODE axioms.**

5.5 Differential Invariants - Equations

We can now put the development of differentials to use. The main proof rule for working with them is called *differential invariants*. Skipping ahead to the proof rule:

$$(\mathrm{dI}_{=}) \quad \frac{\vdash [x' := f(x)](e)' = 0}{e = 0 \vdash [\{x' = f(x)\}]e = 0}$$

This rule intuitively says that if we want to prove an invariant $e = 0$ for the ODE $x' = f(x)$, then all we need to do in the "inductive step" is to prove that its derivative along the solution is always 0. **Note: You can think of differential equations as loops that happen continuously, and of differential invariants as the continuous analogue of loop invariants.**

Note: The rest of these notes give more details than we had time for in recitation.

Let us zoom in on our proof of the time-triggered ping-pong model and see how differential invariants could be used in its proof. We are going to prove:

$$x = x_0, t = 0 \vdash [\{x' = v, t' = 1\}]x - (x_0 + vt) = 0$$

Here is a proof:

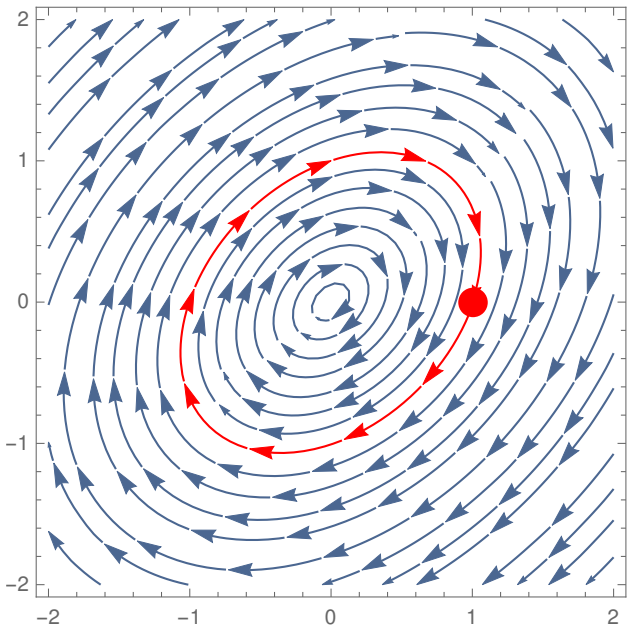
$$\cfrac{\cfrac{\cfrac{\cfrac{\cfrac{}{\vdash v - (0 + 0 + v) = 0}\,*}{\vdash [x' := v][t' := 1]x' - (x_0' + v't + vt') = 0}\,\mathbb{R}}{\vdash [x' := v][t' := 1]\big(x - (x_0 + vt)\big)' = 0}\,[':=]}{x - (x_0 + vt) = 0 \vdash [\{x' = v, t' = 1\}]x - (x_0 + vt) = 0}\,\mathrm{dI}_{=}}{x = x_0, t = 0 \vdash [\{x' = v, t' = 1\}]x - (x_0 + vt) = 0}\,\mathrm{cut}_{,\mathbb{R}}$$

This shows that if $x = x_0, t = 0$ initially, then $x - (x_0 + vt) = 0$ is an invariant of the differential equation. Taking a step back, we have simply solved the ODEs using dI. In fact, KeYmaera X essentially does this reasoning internally, but the reason our proof was much faster is that it bypasses all of the additional manipulation that KeYmaera X has to do for solving an ODEs in general.

However, dI can be used to prove properties of ODEs beyond just their solutions. In class, we already saw how to prove circular invariants, but dI applies beyond just circles. As an example, consider the following system:

$$x' = \frac{-x + 3y}{4}, y' = \frac{-3x + y}{4}$$

Here is a plot of it:



The solutions of this system always trace out a rotated ellipse in the plane. One such solution starting from the point $x = 0, y = 1$ is plotted in red.

It is certainly possible to find the general solution for this differential equation, which will be a complicated expression involving sums of sines and cosines. The point, however, is that we will very often *not* need to know the solution precisely, but rather only need to know properties of the solution.

More concretely, let us suppose that our initial state is the red point above, and we want to show that the solution never reaches an unsafe state where $y > \frac{3}{2}$.

We can now prove this safety property using dI using the fact that the ellipse in red is given by the equation $3(x^2 + y^2) - 2xy - 3 = 0$.

Here is the proof:

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{\vdash 3(2x\frac{-x+3y}{4} + 2y\frac{-3x+y}{4}) - 2(\frac{-x+3y}{4}y + x\frac{-3x+y}{4}) = 0}\,\mathbb{R}}{\vdash [x' := \frac{-x+3y}{4}][y' := \frac{-3x+y}{4}]3(2xx' + 2yy') - 2(x'y + xy') = 0}\,[':=]}{\vdash [x' := \frac{-x+3y}{4}][y' := \frac{-3x+y}{4}]\big(3(x^2 + y^2) - 2xy - 3\big)' = 0}}{3(x^2 + y^2) - 2xy - 3 = 0 \vdash [\{x' = \frac{-x+3y}{4}, y' = \frac{-3x+y}{4}\}]3(x^2 + y^2) - 2xy - 3 = 0}\,\mathrm{dI}_{=}}{x = 1, y = 0 \vdash [\{x' = \frac{-x+3y}{4}, y' = \frac{-3x+y}{4}\}]3(x^2 + y^2) - 2xy - 3 = 0}\,\mathrm{cut}}{x = 1, y = 0 \vdash [\{x' = \frac{-x+3y}{4}, y' = \frac{-3x+y}{4}\}]y \leq \frac{3}{2}}\,\mathrm{M}[\cdot]$$

In the M[·] step, we have strengthened the postcondition because $3(x^2+y^2)-2xy-3 = 0$ implies $y \leq \frac{3}{2}$ (this can be seen from the plot). In the cut step, we proved that the initial point $x = 0, y = 1$ lies on the ellipse.

We then used dI which produced a rather complicated-looking arithmetic term at the end. Fortunately, after a little bit of algebra (or asking Mathematica), all of the terms cancel so that the final premise closes by real arithmetic. Even though the math was a little hairy, it was still far better than solving the ODE and then analyzing its solution. In fact, our

proof only used the fact that the initial point lies on the ellipse. We could have proved this safety property for *all* points on the ellipse.

Exercise 6:
Work through the proof yourself from another initial point, say $x = 1, y = 1$.
Answer: The equation of the associated ellipse is $3(x^2 + y^2) - 2xy - 4 = 0$. Make sure to try the calculation after the $\mathrm{dI}_=$ step yourself. Notice that the calculation after the $\mathrm{dI}_=$ steps are almost exactly identical. Why? (We will see more of this in lectures next week).

Exercise 7:
Solve the ODE explicitly for a closed form solution and compare its complexity to the proof we did above.
Answer: Here is the solution from Mathematica.

$$x(t) = \frac{1}{4}\left(-\sqrt{2}x_0 \sin\left(\frac{t}{\sqrt{2}}\right) + 4x_0 \cos\left(\frac{t}{\sqrt{2}}\right) + 3\sqrt{2}y_0 \sin\left(\frac{t}{\sqrt{2}}\right)\right)$$

$$y(t) = \frac{1}{4}\left(-3\sqrt{2}x_0 \sin\left(\frac{t}{\sqrt{2}}\right) + \sqrt{2}y_0 \sin\left(\frac{t}{\sqrt{2}}\right) + 4y_0 \cos\left(\frac{t}{\sqrt{2}}\right)\right)$$

When $y_0 = 0, x_0 = 1$, $y(t) = -\frac{3\sin\left(\frac{t}{\sqrt{2}}\right)}{2\sqrt{2}}$, which takes on the maximum value $\frac{3}{2\sqrt{2}} \approx 1.06 < 1.5$.