LFCPS Exercise Class 1
Modeling cyber-physical systems

Exercise 1. *Consider the following hybrid programs and compute the reachability relation $[\![\alpha]\!]$ for each one:*

1. $\alpha \equiv ?\,\text{false}$,
2. $\alpha \equiv x := 1; (x := x + 1)^*$,
3. $\alpha \equiv x := 1; (x := x + 1)^*; ?x < 0$,
4. $\alpha \equiv x := 1; (x := x + 1; ?x < 0)^*$.

Exercise 2. *Consider the following hybrid programs and provide an intuitive interpretation for each:*

$$\begin{aligned}
&\alpha \equiv \{x' = v, v' = a \& v \geq 0\} \\
&\beta \equiv \{x' = v, v' = a \& v \geq 0\}\,; ?v = 0 \\
&\gamma \equiv t := 0; \{x' = v, v' = a, t' = 1 \& t \leq T\} \\
&\delta \equiv t := 0; \{x' = v, v' = a, t' = 1 \& t \leq T\}\,; ?(t = T)
\end{aligned}$$

Exercise 3. *Consider the following model of a bouncing ball:*

$$\alpha \equiv \left(\{x' = v,\ v' = -g\}\,;\ v := -cv\right)^*.$$

Here, the variables are defined as follows:

- *x: the height of the ball above the ground.*
- *v: the vertical velocity of the ball, where positive values indicate upward motion and negative values indicate downward motion.*
- *g: the constant acceleration due to gravity, acting downward.*
- *c: the coefficient of restitution, representing the energy retained after each bounce, with $0 < c < 1$.*

Is this a good model for a bouncing ball? Justify your answer by considering the physical behavior and limitations of the model.

Exercise 4. *Consider the following system:*

$$\alpha \equiv \left(\{x' = 1 \ \&\ x \leq 0\} \cup \{x' = -1 \ \&\ x > 0\}\right)^*.$$

Starting with an initial condition $x \leq 0$, analyze which states are reachable under this system. Describe how the evolution of x is constrained by the dynamics specified in α.

Exercise 5. *Consider the following differential equation describing the dynamics of a controlled heating system:*

$$T' = k(T_s - T),$$

where T represents the current temperature, T_s denotes the desired temperature setpoint, and k is a proportional constant related to the system's response rate.

Model a controlled heating system that operates as follows:

- *The heating system turns on and the setpoint is set to 22°C when the temperature falls below 19°C.*
- *The heating system turns off when the temperature exceeds 21°C.*

Exercise 6. *Consider a robot located at a point with coordinates (x, y), facing in a direction represented by the unit vector (d_x, d_y). The robot moves forward in the direction it is facing while simultaneously rotating with angular velocity ω.*

Develop a system of differential equations to describe how the position (x, y) and direction (d_x, d_y) of the robot evolve over time. Construct your model in two steps:

1. *First, model the effect of rotation alone on the direction vector (d_x, d_y).*
2. *Then, consider the translation of the position (x, y) in the current direction (d_x, d_y) of the robot.*

After building these components separately, combine them to form a complete system of differential equations describing the robot's dynamics.

***Bonus**: As an additional challenge, try to extend this model by introducing an acceleration term to describe the robot's increasing linear speed.*

Exercise 7. *1. Can you find a discrete controller* ctrl *and a continuous program* plant *such that the following two HPs α, β have different behaviors?*

$$\alpha \equiv (\text{ctrl}; \text{plant})^* \quad \beta \equiv (\text{ctrl} \cup \text{plant})^*.$$

2. *Can you find a discrete controller ctrl and a continuous program plant with the following two properties: a) The program ctrl has the same behavior as ctrl; ctrl and b) The following two HPs α, β have different behaviors?*

$$\alpha \equiv \text{ctrl}; (\text{ctrl}; \text{plant})^* \quad \beta \equiv \text{ctrl}; (\text{ctrl} \cup \text{plant})^*.$$

Solution Key

Notation. A state ω assigns a real value $\omega(x)$ to every variable x, and ω_x^r is the state that agrees with ω except that x has the value r. We use the semantics of hybrid programs from the lecture:

$$\begin{aligned} [\![x := e]\!] &= \{(\omega, \omega_x^{\omega[\![e]\!]}) : \omega \text{ a state}\} & [\![?Q]\!] &= \{(\omega,\omega) : \omega \in [\![Q]\!]\} \\ [\![\alpha \cup \beta]\!] &= [\![\alpha]\!] \cup [\![\beta]\!] & [\![\alpha;\beta]\!] &= [\![\alpha]\!] \circ [\![\beta]\!] \\ [\![\alpha^*]\!] &= \textstyle\bigcup_{n\in\mathbb{N}} [\![\alpha^n]\!] & \alpha^0 &\equiv ?\,\text{true}, \quad \alpha^{n+1} \equiv \alpha^n;\alpha \end{aligned}$$

where $[\![\alpha]\!] \circ [\![\beta]\!] = \{(\omega,\nu) : (\omega,\mu) \in [\![\alpha]\!] \text{ and } (\mu,\nu) \in [\![\beta]\!] \text{ for some state } \mu\}$. Finally, $(\omega,\nu) \in [\![\{x' = f(x) \;\&\; Q\}]\!]$ iff there is a solution $\varphi\colon [0,r] \to$ States of $x' = f(x)$ of some duration $r \geq 0$ with $\varphi(0) = \omega$, $\varphi(r) = \nu$, and $\varphi(s) \in [\![Q]\!]$ for *all* $s \in [0,r]$, i.e., the evolution domain Q has to hold at the start, throughout, and at the end. The duration r is chosen nondeterministically ($r = 0$ is possible whenever $\omega \in [\![Q]\!]$), and variables without a differential equation keep their values.

Solution to Exercise 1

1. $[\![?\,\text{false}]\!] = \{(\omega,\omega) : \omega \in [\![\text{false}]\!]\} = \emptyset$, because no state satisfies false. The program has no run at all. Consequently, $[?\,\text{false}]P$ is valid and $\langle ?\,\text{false}\rangle P$ is unsatisfiable for every formula P.

2. By induction on n, $[\![(x := x+1)^n]\!] = \{(\omega, \omega_x^{\omega(x)+n}) : \omega \text{ a state}\}$, hence

   $$[\![(x := x+1)^*]\!] = \{(\omega, \omega_x^{\omega(x)+n}) : \omega \text{ a state},\ n \in \mathbb{N}\}.$$

   Composing with $[\![x := 1]\!] = \{(\omega, \omega_x^1) : \omega \text{ a state}\}$ gives

   $$\begin{aligned} [\![\alpha]\!] &= \{(\omega, \omega_x^{1+n}) : \omega \text{ a state},\ n \in \mathbb{N}\} \\ &= \{(\omega, \omega_x^m) : \omega \text{ a state},\ m \in \mathbb{N},\ m \geq 1\}. \end{aligned}$$

   So x ends with an arbitrary positive natural number, independently of its initial value, and all other variables are unchanged. For example, $[\alpha]\, x \geq 1$ and $\langle\alpha\rangle\, x = 5$ are valid.

3. By 2, every final state ν of $x := 1; (x := x+1)^*$ satisfies $\nu(x) \geq 1$, so the final test $?x < 0$ always fails:

   $$[\![\alpha]\!] = \{(\omega,\nu) \in [\![x := 1; (x := x+1)^*]\!] : \nu(x) < 0\} = \emptyset.$$

   Thus α has the same semantics as $?\,\text{false}$.

4. The loop body has the semantics

   $$[\![x := x+1; ?x < 0]\!] = \{(\omega, \omega_x^{\omega(x)+1}) : \omega \text{ a state with } \omega(x)+1 < 0\}.$$

   After $x := 1$ we have $x = 1$ and $1 + 1 < 0$ is false, so the loop body has no transition from such a state. Hence no run performs $n \geq 1$ iterations, and only $n = 0$ (that is, $?\,\text{true}$) contributes:

   $$[\![\alpha]\!] = \{(\omega, \omega_x^1) : \omega \text{ a state}\} = [\![x := 1]\!].$$

   Compare with 3: a failing test after the loop discards the whole run, whereas a failing test inside the loop body only rules out the runs that perform one more iteration. Since a loop may always stop after zero iterations, the run of $x := 1$ survives.

Solution to Exercise 2

In all four programs, think of x as the position, v as the velocity and a as the acceleration of a car on a straight road. Since a and T have no differential equation, they are constant during the continuous evolution. The solution of $x' = v$, $v' = a$ from initial values x, v is $x(s) = x + vs + \frac{a}{2}s^2$, $v(s) = v + as$.

α: The car follows the dynamics for a nondeterministically chosen duration $r \geq 0$, as long as the velocity is non-negative *throughout*, i.e., the car never drives backwards:

$$[\![\alpha]\!] = \{(\omega, \nu) : \text{for some } r \geq 0,\ \omega(v) \geq 0,\ \omega(v) + \omega(a)r \geq 0, \text{ and } \nu = \omega \text{ except}$$
$$\nu(x) = \omega(x) + \omega(v)r + \tfrac{\omega(a)}{2}r^2,\ \nu(v) = \omega(v) + \omega(a)r\}$$

(since $v(s)$ is linear in s, it is non-negative on all of $[0, r]$ iff it is non-negative at $s = 0$ and at $s = r$). If $v < 0$ initially, α has no run at all. If $a \geq 0$, the evolution can last arbitrarily long. If $a < 0$ (braking), it can last at most until standstill, i.e., $r \leq -\frac{v}{a}$. The domain does not make the car stop at $v = 0$; it only forbids evolving any further, and α may just as well stop at any earlier time (including $r = 0$).

β: Like α, but the test $?v = 0$ only keeps the runs that end with velocity 0: the car drives (brakes) *until it comes to a full stop*. Domain $v \geq 0$ followed by the test $?v = 0$ is the typical pattern for "follow the ODE until $v = 0$". If initially $v \geq 0$ and $a < 0$, there is exactly one run, of duration $r = -\frac{v}{a}$, ending with velocity 0 at the stopping position $x - \frac{v^2}{2a}$. If $a \geq 0$, the test can only succeed when $v = 0$ initially, in which case the final state equals the initial state. If $v > 0$ and $a \geq 0$, there is no run (the car cannot come to a stop without braking).

γ: t is a clock: it is reset to 0 and evolves with $t' = 1$, so it measures the duration of the evolution. The domain $t \leq T$ allows exactly the durations r with $0 \leq r \leq T$: the car follows the dynamics for *at most* T time units (there is no run if $T < 0$). The velocity is unconstrained, so the car may also drive backwards. Inside a control loop $(\text{ctrl}; \gamma)^*$ this models a time-triggered controller that runs again after at most T time units (maximal sampling period T).

δ: Like γ, but the test $?t = T$ only keeps the runs of duration *exactly* T. For $T \geq 0$, δ is deterministic: from initial values x, v its unique final state has position $x + vT + \frac{a}{2}T^2$, velocity $v + aT$, and $t = T$ (there is no run if $T < 0$). Inside a control loop this models periodic sampling with exact period T.

Solution to Exercise 3

No. The model captures the flight under gravity (the ODE), the loss of energy at a bounce ($v := -cv$ with $0 < c < 1$), and repeated bounces (the loop), but it does not restrict *when* each of these may happen:

1. *No evolution domain.* The ODE $\{x' = v, v' = -g\}$ may run for an arbitrarily long time, so the ball can fall through the ground. For instance, with $g = 10$, $c = \frac{1}{2}$ and initial values $x = 5$, $v = 0$, following the ODE for 2 time units leads to $x = 5 - \frac{10}{2} \cdot 2^2 = -15$. Hence $x = H \wedge H \geq 0 \wedge v = 0 \wedge g > 0 \wedge 0 < c \wedge c < 1 \rightarrow [\alpha]\, x \geq 0$ is not valid. Fix: add the evolution domain $x \geq 0$.

2. *The bounce is not tied to the ground.* $v := -cv$ is executed after every continuous evolution, whatever its duration, so the ball can also bounce in mid-air (or below the ground). With the values above, flying for $\frac{1}{2}$ time unit gives $x = 3.75$ and

$v = -5$, and the bounce then reverses the falling ball at height 3.75 to $v = 2.5$. Fix: bounce only on the ground, $?x = 0; v := -cv$, and do nothing otherwise, $?x \neq 0$.

A better model is

$$\alpha_{\text{bb}} \equiv \big(\{x' = v,\, v' = -g \;\&\; x \geq 0\};\; \big((?x = 0;\, v := -cv) \cup ?x \neq 0\big)\big)^*,$$

where the second part implements "if $x = 0$ then $v := -cv$". Now the ball cannot fly below the ground, and it bounces exactly when it is on the ground. Assumptions such as $g > 0$ and the initial state are not part of the hybrid program; they go into the precondition of the dL formula expressing the intended property, e.g.,

$$x = H \wedge H \geq 0 \wedge v = 0 \wedge g > 0 \wedge 0 < c \wedge c < 1 \;\rightarrow\; [\alpha_{\text{bb}}]\,(0 \leq x \wedge x \leq H),$$

which is valid (unlike the corresponding formula for the original α).

Remaining limitations are deliberate abstractions: the ball is a point mass, the bounce is instantaneous, there is no air resistance, and g and c are constant. Two further subtleties of α_{bb}: (i) since the ODE may run for duration 0, a ball on the ground can bounce twice in a row without time passing (e.g., with $c = \frac{1}{2}$, $v = -5$ becomes 2.5 and then -1.25); this only removes energy and is harmless for the safety property above, but it can be excluded by bouncing only if $x = 0 \wedge v \leq 0$, i.e., using $(?(x = 0 \wedge v \leq 0);\, v := -cv) \cup ?\neg(x = 0 \wedge v \leq 0)$ instead; (ii) the bounce times of a real ball accumulate (Zeno behavior), so the ball comes to rest after finite time and infinitely many bounces, whereas every run of a hybrid program has only finitely many loop iterations. Hence the model only describes the ball before that accumulation point, which again does not affect safety properties like $0 \leq x \leq H$.

Solution to Exercise 4

The two continuous programs have the semantics

$$\begin{aligned}
[\![\{x' = 1 \;\&\; x \leq 0\}]\!] &= \{(\omega, \omega_x^{\omega(x)+r}) : r \geq 0 \text{ and } \omega(x) + s \leq 0 \text{ for all } s \in [0, r]\} \\
&= \{(\omega, \omega_x^{\omega(x)+r}) : r \geq 0 \text{ and } \omega(x) + r \leq 0\}, \\
[\![\{x' = -1 \;\&\; x > 0\}]\!] &= \{(\omega, \omega_x^{\omega(x)-r}) : r \geq 0 \text{ and } \omega(x) - r > 0\}.
\end{aligned}$$

Let ω be an initial state with $x_0 = \omega(x) \leq 0$.

- The second branch can never be executed from a state with $x \leq 0$, since its domain $x > 0$ already has to hold at the beginning of the evolution.
- The first branch increases x at rate 1, but only as long as $x \leq 0$ holds throughout, *including at the end*. From a state with $x \leq 0$, it reaches exactly the values in $[x, 0]$; it can reach 0 but never exceed it.

Hence, every iteration of the loop starts and ends with $x \leq 0$ and never decreases x, and every value in $[x_0, 0]$ is already reachable with one iteration. The reachable states are exactly those in which x has a value between x_0 and 0 and all other variables are unchanged:

$$\{\nu : (\omega, \nu) \in [\![\alpha]\!]\} = \{\omega_x^r : x_0 \leq r \leq 0\}.$$

In dL, with a fresh variable x_0 that does not occur in α, the formula $x = x_0 \wedge x_0 \leq 0 \rightarrow [\alpha]\,(x_0 \leq x \wedge x \leq 0)$ is valid, and so is $x = x_0 \wedge x_0 \leq b \wedge b \leq 0 \rightarrow \langle\alpha\rangle\, x = b$ (with b also fresh): the bounds are exact.

In particular, the system never switches to $x' = -1$, although one might expect it to oscillate around 0: the first ODE can never make x positive, and the second ODE can never start with $x \leq 0$. Once $x = 0$, only evolutions of duration 0 are possible ($x' = 1$ would leave $x \leq 0$ immediately, and $x' = -1$ requires $x > 0$), so time cannot progress any further. Symmetrically, from an initial value $x_0 > 0$ only the second branch is possible, which can bring x arbitrarily close to 0 but never to 0 because the domain $x > 0$ is open; then the reachable values are exactly $(0, x_0]$.

Solution to Exercise 5

Modeling decisions:

- dL only has real-valued variables, so we use $h = 1$ for "heater on" and $h = 0$ for "heater off".
- When the heater is off, no heat is supplied and the room relaxes towards the ambient temperature T_e (Newton's law of cooling). This is the same differential equation with setpoint T_e, so turning the heater off sets $T_s := T_e$. We assume $k > 0$ and $T_e < 19$ (otherwise the room never cools down to 19 and the heater is never needed).
- The controller must be able to react when a threshold is reached, so the evolution domain stops the continuous evolution at $T = 21$ while the heater is on and at $T = 19$ while it is off (event-triggered control).

$$\begin{aligned}
\text{ctrl} &\equiv (?T \leq 19;\ h := 1;\ T_s := 22) \cup (?T \geq 21;\ h := 0;\ T_s := T_e) \\
&\quad \cup\ ?(19 < T \wedge T < 21) \\
\text{plant} &\equiv \{T' = k(T_s - T)\ \&\ (h = 1 \wedge T \leq 21) \vee (h = 0 \wedge T \geq 19)\} \\
\text{heat} &\equiv (\text{ctrl}; \text{plant})^*
\end{aligned}$$

The three tests of ctrl are mutually exclusive and cover all temperatures, so the controller is deterministic and never blocks. Between 19 and 21 the heater keeps its current state (hysteresis). If initially $h \in \{0, 1\}$, the evolution domain of plant is always satisfied after ctrl, so time can always progress.

Why the tests are $T \leq 19$ and $T \geq 21$ instead of $T < 19$ and $T > 21$: the domain makes the evolution stop at the threshold itself, and there is no first moment at which $T > 21$ holds (the set of such times is open), so "exceeds 21" has to be detected at the boundary $T = 21$. With the strict tests $?T > 21$ and $?T < 19$, the system would get stuck at $T = 21$: the plant could only run for duration 0 and the controller would never switch the heater off. Then no time could pass, so the model would not describe any behavior after that point and a safety proof would say nothing about it, which is a modeling error. Without the domains, on the other hand, the controller could miss the thresholds: the plant could keep heating towards 22 or cool all the way towards T_e.

The intended property can now be stated in dL and is valid:

$$k > 0 \wedge T_e < 19 \wedge 19 \leq T \wedge T \leq 21 \wedge \big((h = 1 \wedge T_s = 22) \vee (h = 0 \wedge T_s = T_e)\big) \rightarrow [\text{heat}]\,(19 \leq T \wedge T \leq 21).$$

The setpoint 22 lying strictly above the threshold 21 is important: T only approaches T_s asymptotically, so with a setpoint of at most 21 the heater would never reach 21 and never turn off.

Alternatively, a time-triggered model takes the thresholds literally and lets the controller run at least every $\varepsilon > 0$ time units:

$$\begin{aligned}\text{ctrl} &\equiv (?T < 19;\ h := 1;\ T_s := 22) \cup (?T > 21;\ h := 0;\ T_s := T_e)\\ &\quad \cup\ ?(19 \le T \wedge T \le 21)\\ \text{plant} &\equiv t := 0;\ \{T' = k(T_s - T),\ t' = 1\ \&\ t \le \varepsilon\}\end{aligned}$$

In this model the temperature may overshoot the thresholds between two control decisions.

Solution to Exercise 6

Terms of dL are polynomials, so sin and cos are not available, and an orientation angle θ with $x' = v\cos\theta$ cannot be used. This is why the direction is represented by the unit vector (d_x, d_y), which takes the role of $(\cos\theta, \sin\theta)$.

1. *Rotation.* When the vector (d_x, d_y) rotates with angular velocity ω, its derivative is perpendicular to it, namely (d_x, d_y) rotated by 90° counterclockwise and scaled by ω:
$$\{d_x' = -\omega d_y,\ d_y' = \omega d_x\}$$
(counterclockwise rotation for $\omega > 0$). Sanity checks: $(d_x^2 + d_y^2)' = 2d_x(-\omega d_y) + 2d_y(\omega d_x) = 0$, so the direction stays a unit vector, i.e., the following formula is valid:
$$d_x^2 + d_y^2 = 1 \to [\{d_x' = -\omega d_y,\ d_y' = \omega d_x\}]\, d_x^2 + d_y^2 = 1.$$
Moreover, from $(d_x, d_y) = (1, 0)$ the solution is $(\cos(\omega t), \sin(\omega t))$, a rotation by the angle ωt.

2. *Translation.* Moving forward with linear speed v in the current direction:
$$\{x' = vd_x,\ y' = vd_y\}.$$
Since (d_x, d_y) is a unit vector, the robot moves with speed $|v|$ (use $v = 1$ if the robot should move with unit speed).

Both effects happen simultaneously, so they are combined into one system of differential equations (a sequential composition would instead alternate between turning on the spot and driving straight):

$$\{x' = vd_x,\ y' = vd_y,\ d_x' = -\omega d_y,\ d_y' = \omega d_x\}.$$

Since v and ω have no differential equations, they are constant during the evolution: for $\omega \neq 0$ the robot drives along a circle of radius $\frac{|v|}{|\omega|}$, and for $\omega = 0$ along a straight line. A controller can change ω between continuous evolutions, e.g.,

$$(\omega := *;\ \{x' = vd_x,\ y' = vd_y,\ d_x' = -\omega d_y,\ d_y' = \omega d_x\})^*.$$

Bonus: add a differential equation for the linear speed with constant acceleration a:

$$\{x' = vd_x,\ y' = vd_y,\ v' = a,\ d_x' = -\omega d_y,\ d_y' = \omega d_x\ \&\ v \ge 0\}.$$

For $a > 0$ the speed increases linearly, $v(t) = v + at$, and for $\omega \neq 0$ the robot drives along an outward spiral (the radius $\frac{v}{|\omega|}$ grows). The domain $v \ge 0$ is only needed if braking ($a < 0$) is allowed as well; it prevents the robot from driving backwards.

Solution to Exercise 7

Two HPs have the same behavior iff they have the same reachability relation. In general $[\![\alpha]\!] \subseteq [\![\beta]\!]$ holds in both parts, because one iteration ctrl; plant of α can be mimicked by two iterations of β (first ctrl, then plant). Thus $[\beta]P \to [\alpha]P$ is valid for every P, and a difference can only come from the additional freedom of β: it can (i) run plant before any ctrl, (ii) run ctrl several times in a row, (iii) run plant several times in a row, and (iv) end with ctrl.

1. Yes, e.g., ctrl $\equiv x := 0$ and plant $\equiv \{x' = 1\}$. From an initial value x_0, α reaches exactly the values $x \in \{x_0\} \cup [0, \infty)$: with zero iterations $x = x_0$, and otherwise the last iteration resets x to 0 before the plant increases it. On the other hand, β reaches all $x \in [x_0, \infty) \cup [0, \infty)$, because it can run plant without running ctrl first. For $x_0 = -1$, the HP β reaches $x = -\frac{1}{2}$ (run plant for $\frac{1}{2}$ time unit), but α does not. In dL: $x = -1 \to [\alpha]\,(x = -1 \lor x \geq 0)$ is valid, but $x = -1 \to [\beta]\,(x = -1 \lor x \geq 0)$ is not.

2. Yes. First note that the example from 1 no longer works: $x := 0; x := 0$ has the same behavior as $x := 0$, but now both α and β reach exactly the states with $x \geq 0$ (all other variables unchanged). The leading ctrl removes difference (i), and property a) removes difference (ii). Difference (iii) does not help either, because for a differential equation $[\![\text{plant; plant}]\!] = [\![\text{plant}]\!]$ (two consecutive solutions can be glued into one, and one solution can be split into two). So the only new runs of β are those that end with ctrl after at least one plant. Such a run can be mimicked by α by appending a run of plant of duration 0, unless the evolution domain of plant is false after the final ctrl. Hence ctrl must be able to leave the evolution domain of plant, for example:

$$\text{ctrl} \equiv u := 0 \cup u := 1 \qquad \text{plant} \equiv \{t' = 1 \,\&\, u = 1\}$$

   a) $[\![\text{ctrl; ctrl}]\!] = \{(\omega, \omega_u^0), (\omega, \omega_u^1) : \omega \text{ a state}\} = [\![\text{ctrl}]\!]$, since the second choice overwrites the first.

   b) Start in a state with $t = 0$. Every run of α either has zero loop iterations (then $t = 0$) or ends with plant, whose domain forces $u = 1$ at the end. Hence $t = 0 \to [\alpha]\,(u = 0 \to t = 0)$ is valid. However, β can run $u := 1$, then plant for 1 time unit, and then $u := 0$, so $t = 0 \to \langle\beta\rangle\,(u = 0 \land t = 1)$ is valid and $t = 0 \to [\beta]\,(u = 0 \to t = 0)$ is not. Intuitively, $u = 1$ means that the controller permits the plant to evolve. In α every control decision is immediately followed by the plant, so the final decision can only be $u := 0$ if no time has passed at all, whereas β may end with the decision $u := 0$ after time has passed.