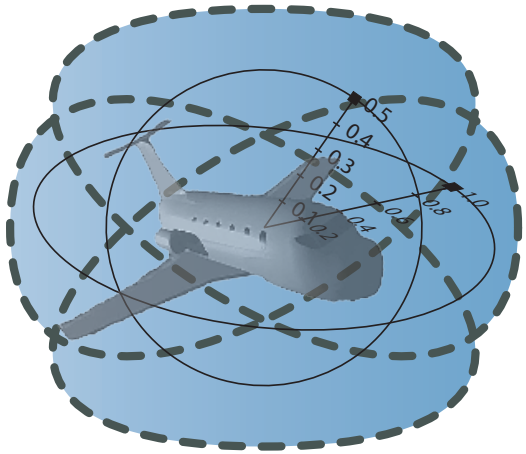
15-819/18-879: Logical Analysis of Hybrid Systems

12: Hybrid Program Semantics

André Platzer

aplatzer@cs.cmu.edu
Carnegie Mellon University, Pittsburgh, PA



Outline

Outline

Differential Dynamic Logic d$\mathcal{L}$: Syntax

Definition (Hybrid program α)

$x' = f(x)$	(continuous evolution)	
$x := f(x)$	(discrete jump)	jump & test
$\texttt{if}(\chi)\,\alpha\,\texttt{else}\,\beta$	(conditional execution)	jump & test
$\alpha;\beta$	(seq. composition)	Kleene algebra
$\alpha \cup \beta$	(nondet. choice)	Kleene algebra
α^*	(nondet. repetition)	Kleene algebra

Differential Dynamic Logic dℒ: Syntax

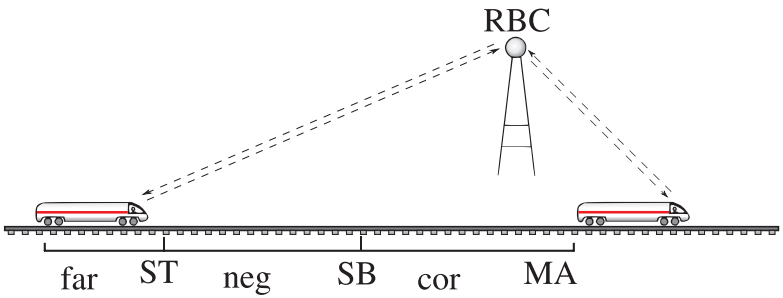
Definition (Hybrid program α)

$x' = f(x)$	(continuous evolution)	
$x := f(x)$	(discrete jump)	jump & test
$\texttt{if}(\chi)\,\alpha\,\texttt{else}\,\beta$	(conditional execution)	jump & test
$\alpha;\beta$	(seq. composition)	Kleene algebra
$\alpha \cup \beta$	(nondet. choice)	Kleene algebra
α^*	(nondet. repetition)	Kleene algebra

$$ETCS \equiv (ctrl\,;drive)^*$$

$$ctrl \equiv \texttt{if}\ (MA - z < SB)\ \texttt{then}\ a := -b$$
$$\texttt{else}\ a := \ldots$$

$$drive \equiv z'' = a$$
$$\wedge\, v \geq 0 \wedge \tau \leq \varepsilon$$



Differential Dynamic Logic d$\mathcal{L}$: Syntax

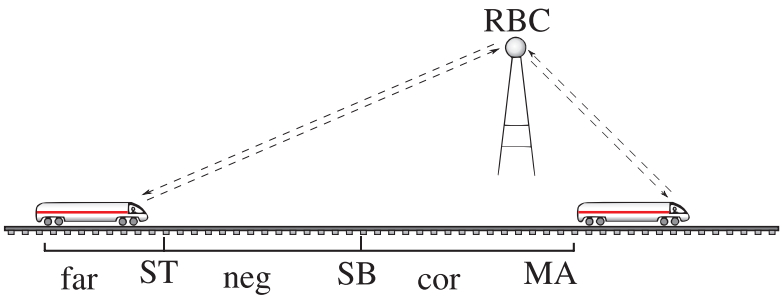
Definition (Hybrid program α)

$x' = f(x)$	(continuous evolution)	
$x := f(x)$	(discrete jump)	jump & test
$\texttt{if}(\chi)\,\alpha\,\texttt{else}\,\beta$	(conditional execution)	jump & test
$\alpha;\beta$	(seq. composition)	Kleene algebra
$\alpha \cup \beta$	(nondet. choice)	Kleene algebra
α^*	(nondet. repetition)	Kleene algebra

$$ETCS \equiv (ctrl\,;drive)^*$$

$$ctrl \equiv \texttt{if}\ (MA - z < SB)\ \texttt{then}\ a := -b$$
$$\texttt{else}\ a := \ldots$$

$$drive \equiv \tau := 0; z' = v, v' = a, \tau' = 1$$
$$\wedge\, v \geq 0 \wedge \tau \leq \varepsilon$$



Differential Dynamic Logic dℒ: Syntax

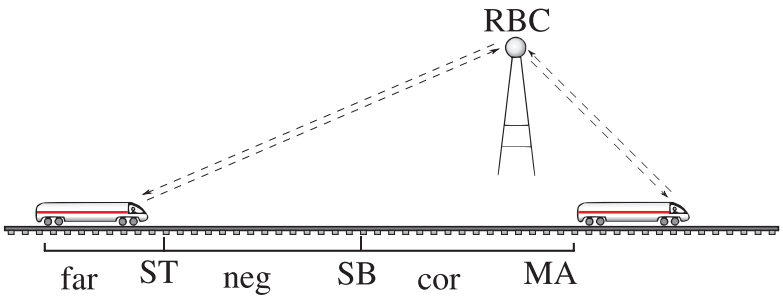
Definition (Hybrid program α)

$x' = f(x) \wedge \chi$	(continuous evolution)	
$x := f(x)$	(discrete jump)	jump & test
$\texttt{if}(\chi)\,\alpha\,\texttt{else}\,\beta$	(conditional execution)	jump & test
$\alpha;\beta$	(seq. composition)	Kleene algebra
$\alpha \cup \beta$	(nondet. choice)	Kleene algebra
α^*	(nondet. repetition)	Kleene algebra

$$ETCS \equiv (ctrl\,;drive)^*$$

$$ctrl \equiv \texttt{if}\ (MA - z < SB)\ \texttt{then}\ a := -b$$
$$\texttt{else}\ a := \ldots$$

$$drive \equiv \tau := 0; z' = v, v' = a, \tau' = 1$$
$$\wedge\, v \geq 0 \wedge \tau \leq \varepsilon$$

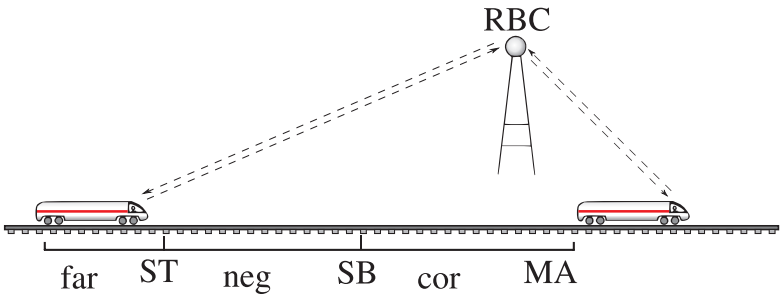


Differential Dynamic Logic d$\mathcal{L}$: Syntax

$$ETCS \equiv (ctrl\,;drive)^*$$

$$ctrl \equiv \texttt{if } (MA - z < SB) \texttt{ then } a := -b \texttt{ else } a := \ldots$$

$$drive \equiv \tau := 0; z' = v, v' = a, \tau' = 1 \wedge v \geq 0 \wedge \tau \leq \varepsilon$$



States in Hybrid Systems

What is a state of a hybrid program?

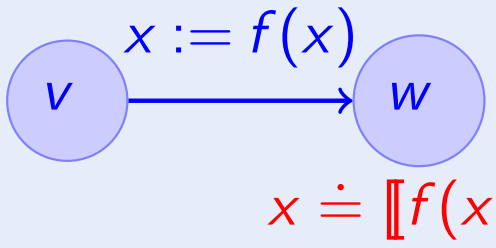
States in Hybrid Systems

What is a state of a hybrid program?

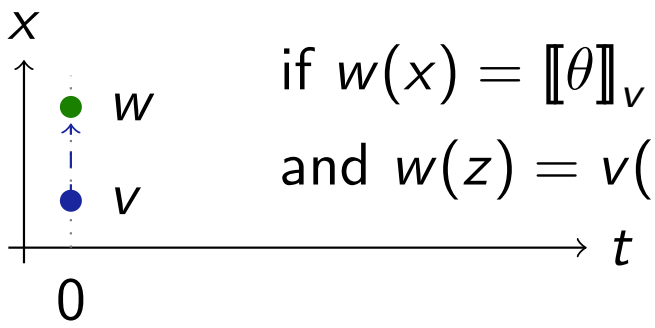
Definition (Kripke state)

$v : V \to \mathbb{R}$ with set of variables V

Definition (Hybrid programs α: transition semantics)

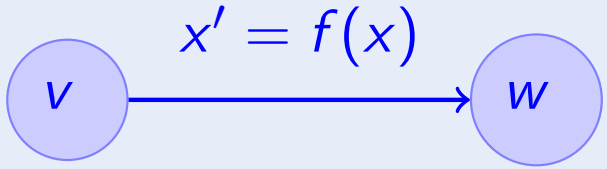


▸ Details ▸▸

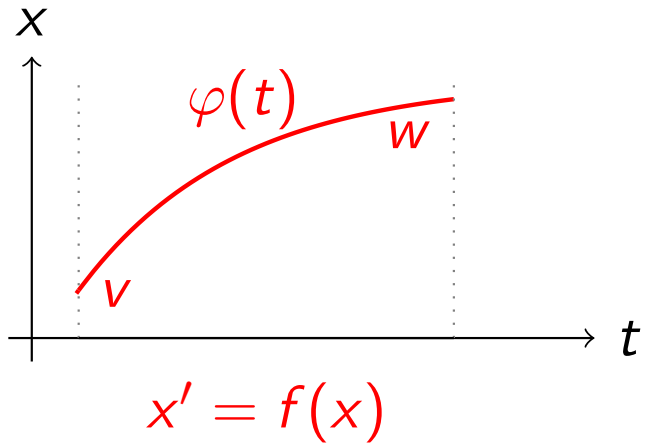


if $w(x) = [\![\theta]\!]_v$
and $w(z) = v(z)$ for $z \neq x$

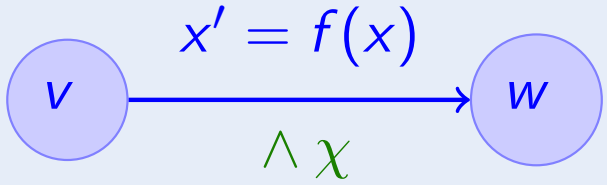
Definition (Hybrid programs α: transition semantics)



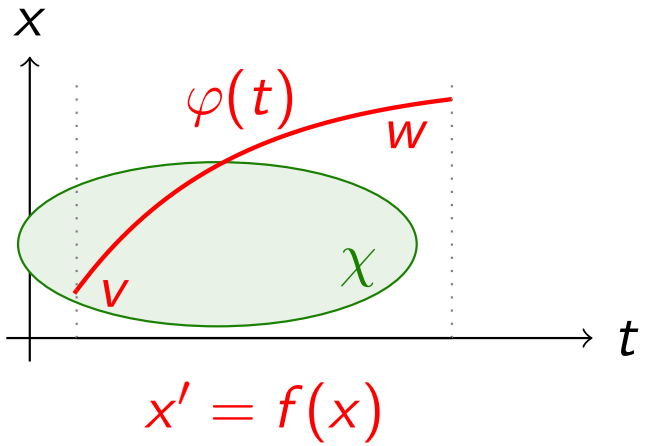
▸ Details ▸▸



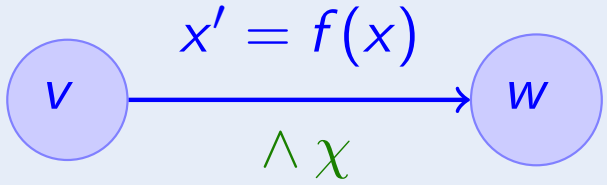
Definition (Hybrid programs α: transition semantics)



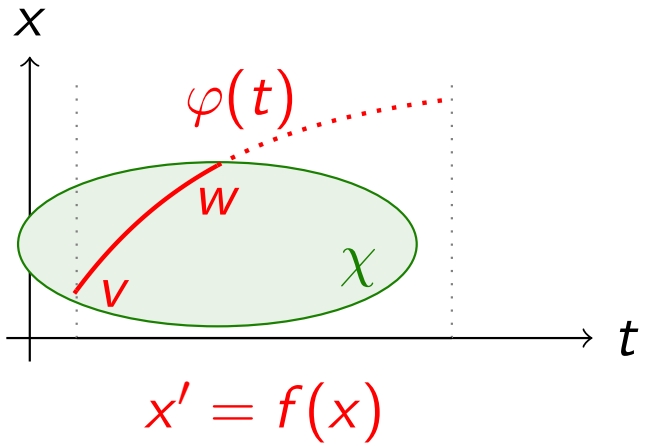
▸ Details ▸▸



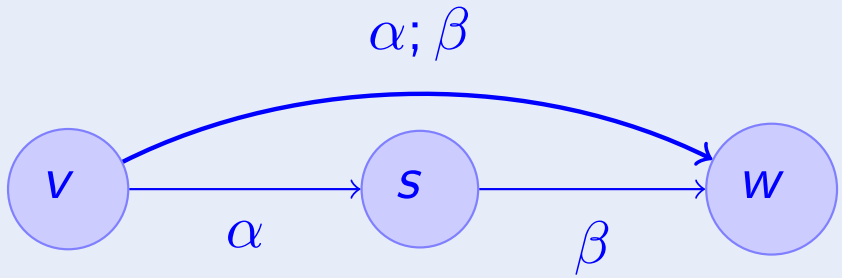
Definition (Hybrid programs α: transition semantics)



▸ Details ▸▸

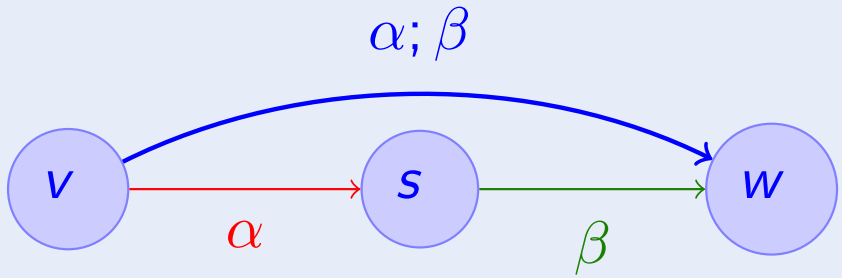


Definition (Hybrid programs α: transition semantics)

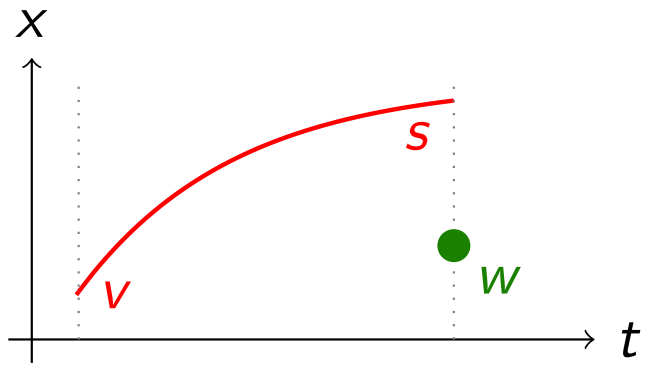


▸ Details ⏩

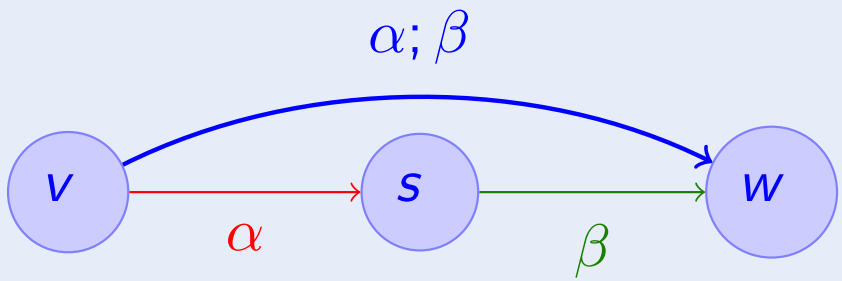
Definition (Hybrid programs α: transition semantics)



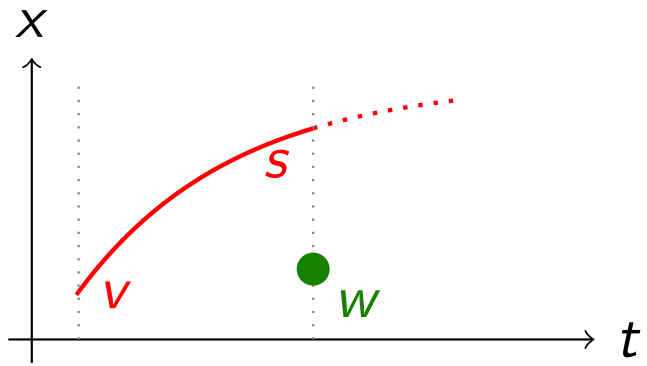
▸ Details ▸▸



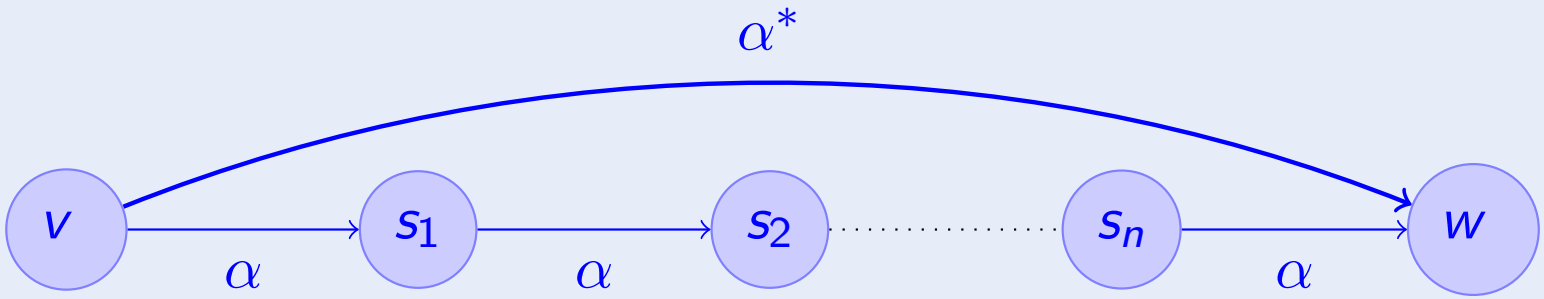
Definition (Hybrid programs α: transition semantics)



▸ Details ▸▸

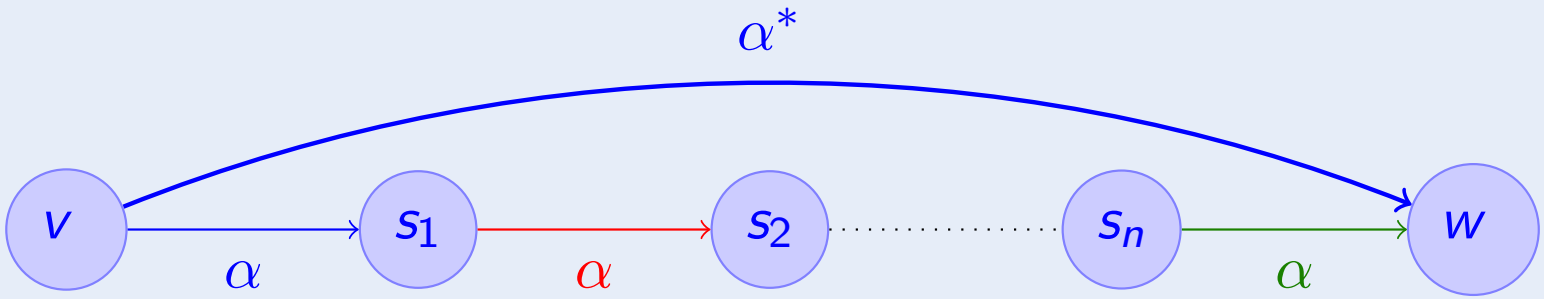


Definition (Hybrid programs α: transition semantics)

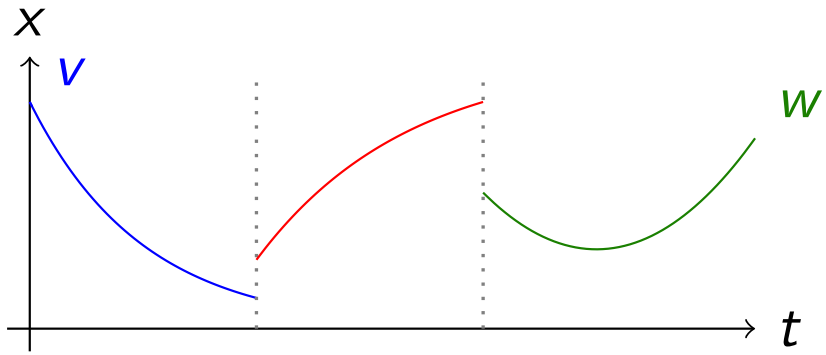


Details

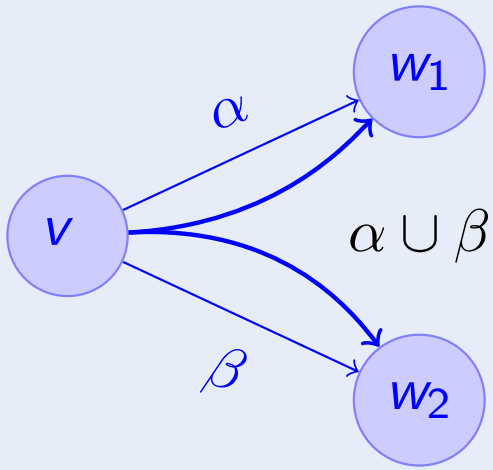
Definition (Hybrid programs α: transition semantics)



▸ Details ▸▸

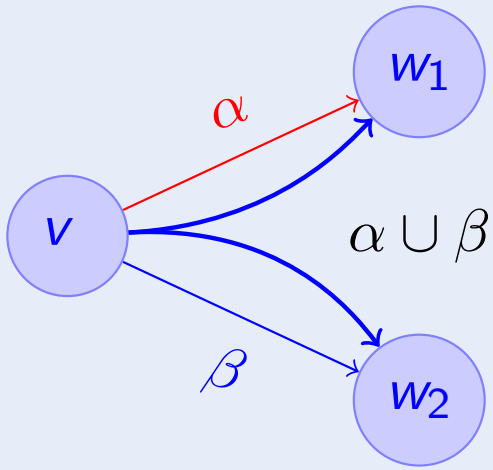


Definition (Hybrid programs α: transition semantics)

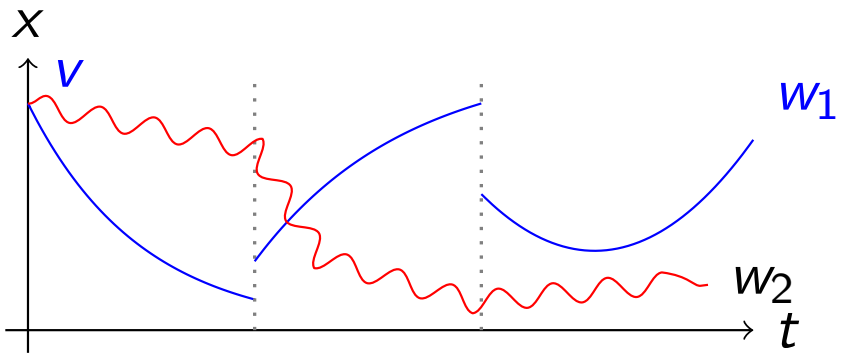


▸ Details ▸▸

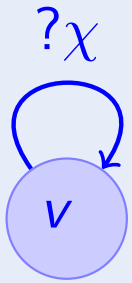
Definition (Hybrid programs α: transition semantics)



▸ Details ▸▸

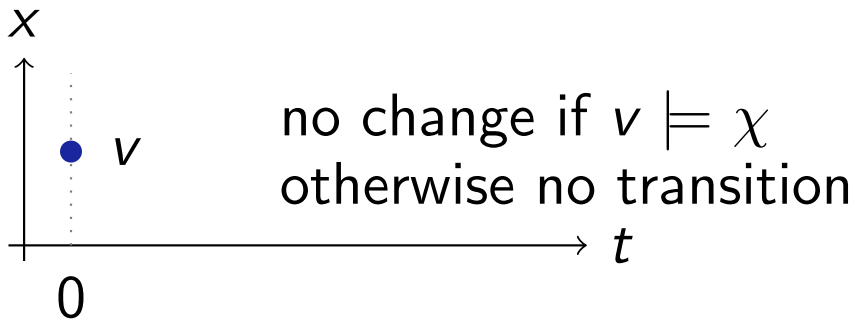


Definition (Hybrid programs α: transition semantics)



if $v \models \chi$

Details

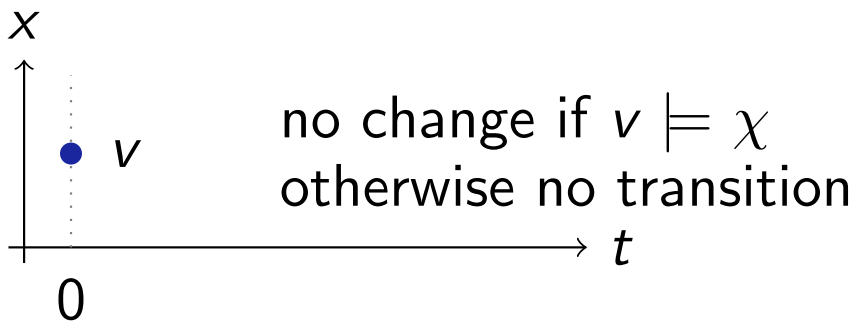


Differential Dynamic Logic d$\mathcal{L}$: Transition Semantics

Definition (Hybrid programs α: transition semantics)

v if $v \not\models \chi$

Details



Hybrid Programs: Formal Semantics

Definition (Hybrid programs α)

$$\begin{aligned}
\rho(x' = f(x)) &= \{(\varphi(0), \varphi(r)) : \varphi \models x' = f(x) \text{ for duration } r\} \\
(v, w) \in \rho(x := \theta) &:\Longleftrightarrow w = v[x \mapsto [\![\theta]\!]_v] \\
\rho(?\chi) &= \{(v, v) : v \models \chi\} \\
\rho(\alpha \cup \beta) &= \rho(\alpha) \cup \rho(\beta) \\
\rho(\alpha; \beta) &= \rho(\alpha) \circ \rho(\beta)
\end{aligned}$$

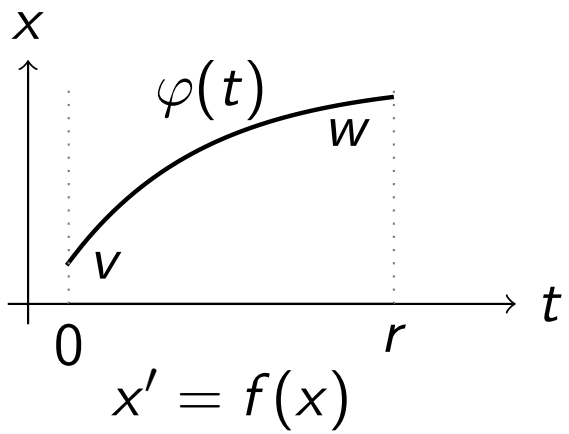
$$(v, w) \in \rho(\alpha^*) :\Longleftrightarrow \text{there is } v \xrightarrow{\rho(\alpha)} v_1 \xrightarrow{\rho(\alpha)} v_2 \cdots\cdots \xrightarrow{\rho(\alpha)} w$$

Definition (Hybrid programs α)

$$\rho(x' = f(x)) \quad = \quad \{(\varphi(0), \varphi(r)) \ : \ \varphi \models x' = f(x) \text{ for duration } r\}$$

$$\text{with } [\![x']\!]_{\varphi(\zeta)} = \frac{\mathrm{d}\varphi(t)(x)}{\mathrm{d}t}(\zeta)$$

- there is $\varphi : [0, r] \to \text{States}$ "with $\varphi(0) = v, \varphi(r) = w$"
- $[\![x]\!]_{\varphi(\zeta)}$ is continuous in ζ on $[0, r]$
- $\frac{\mathrm{d}\,[\![x]\!]_{\varphi(t)}}{\mathrm{d}t}(\zeta) = [\![f(x)]\!]_{\varphi(\zeta)}$ for $\zeta \in (0, r)$
- $[\![y]\!]_{\varphi(\zeta)} = [\![y]\!]_v$ otherwise

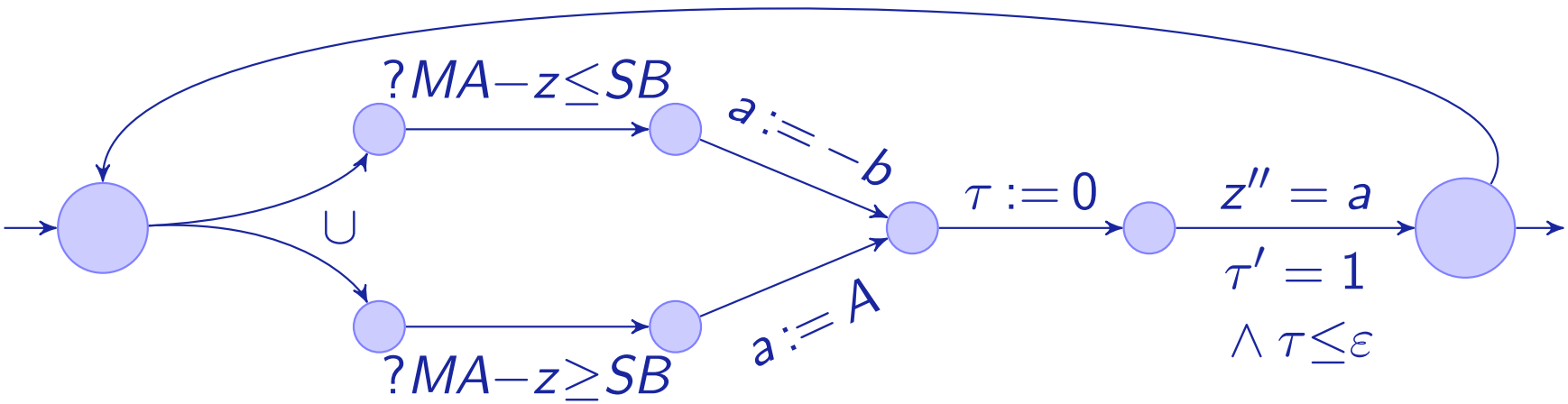


Branching Transitions in Hybrid Programs: ETCS

$$
\begin{aligned}
\text{system} &\equiv (cor\,;drive)^* \\
cor &\equiv (?MA - z \leq SB; a := -b) \cup (?MA - z \geq SB; a := A) \\
drive &\equiv \tau := 0; (z' = v, v' = a, \tau' = 1 \wedge v \geq 0 \wedge \tau \leq \varepsilon)
\end{aligned}
$$

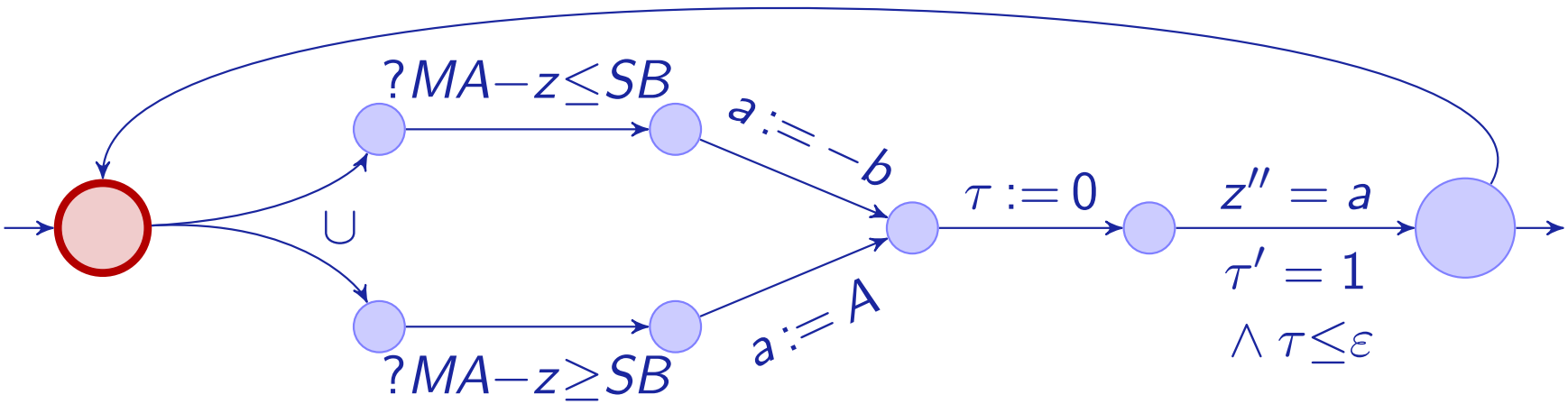
Branching Transitions in Hybrid Programs: ETCS

$$
\begin{aligned}
\text{system} &\equiv (cor\,;drive)^* \\
cor &\equiv (?MA - z \leq SB; a := -b) \cup (?MA - z \geq SB; a := A) \\
drive &\equiv \tau := 0; (z' = v, v' = a, \tau' = 1 \wedge v \geq 0 \wedge \tau \leq \varepsilon)
\end{aligned}
$$



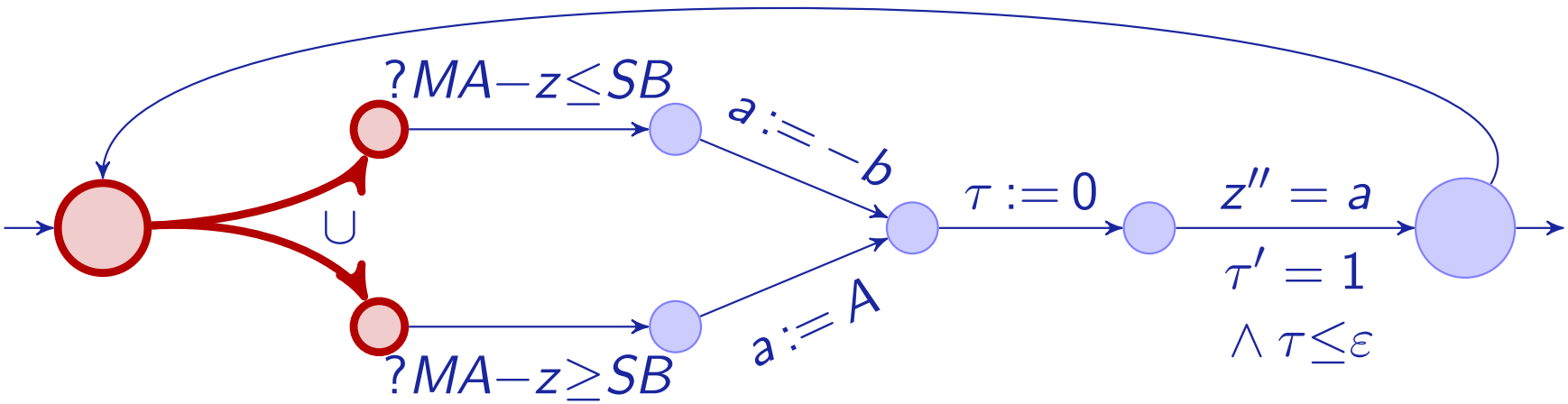
Branching Transitions in Hybrid Programs: ETCS

$$
\begin{aligned}
\text{system} &\equiv (cor\,;drive)^* \\
cor &\equiv (?MA - z \leq SB; a := -b) \cup (?MA - z \geq SB; a := A) \\
drive &\equiv \tau := 0; (z' = v, v' = a, \tau' = 1 \wedge v \geq 0 \wedge \tau \leq \varepsilon)
\end{aligned}
$$



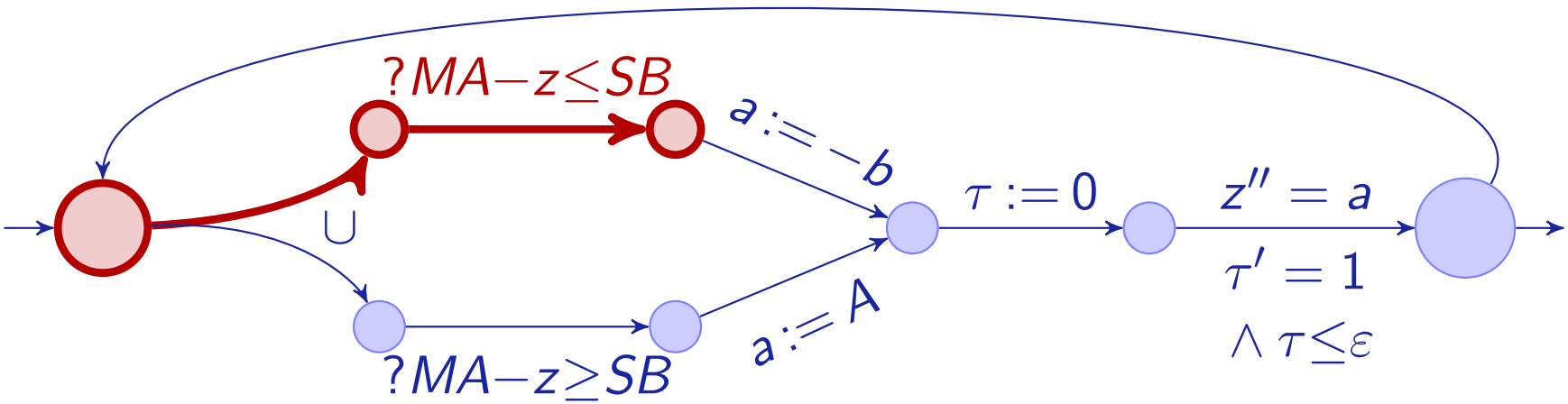
Branching Transitions in Hybrid Programs: ETCS

$$\begin{aligned} \text{system} &\equiv (\mathit{cor}\,;\mathit{drive})^* \\ \mathit{cor} &\equiv (?MA - z \leq SB; a := -b) \cup (?MA - z \geq SB; a := A) \\ \mathit{drive} &\equiv \tau := 0; (z' = v, v' = a, \tau' = 1 \wedge v \geq 0 \wedge \tau \leq \varepsilon) \end{aligned}$$



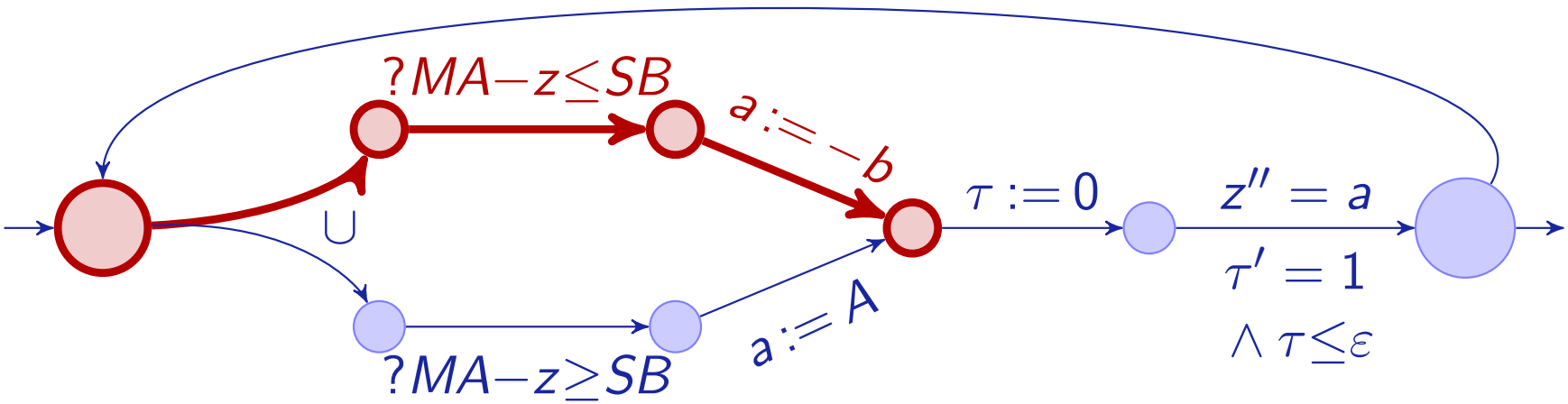
Branching Transitions in Hybrid Programs: ETCS

$$\begin{aligned} \text{system} &\equiv (\mathit{cor}\,;\mathit{drive})^* \\ \mathit{cor} &\equiv (?MA - z \leq SB; a := -b) \cup (?MA - z \geq SB; a := A) \\ \mathit{drive} &\equiv \tau := 0; (z' = v, v' = a, \tau' = 1 \wedge v \geq 0 \wedge \tau \leq \varepsilon) \end{aligned}$$



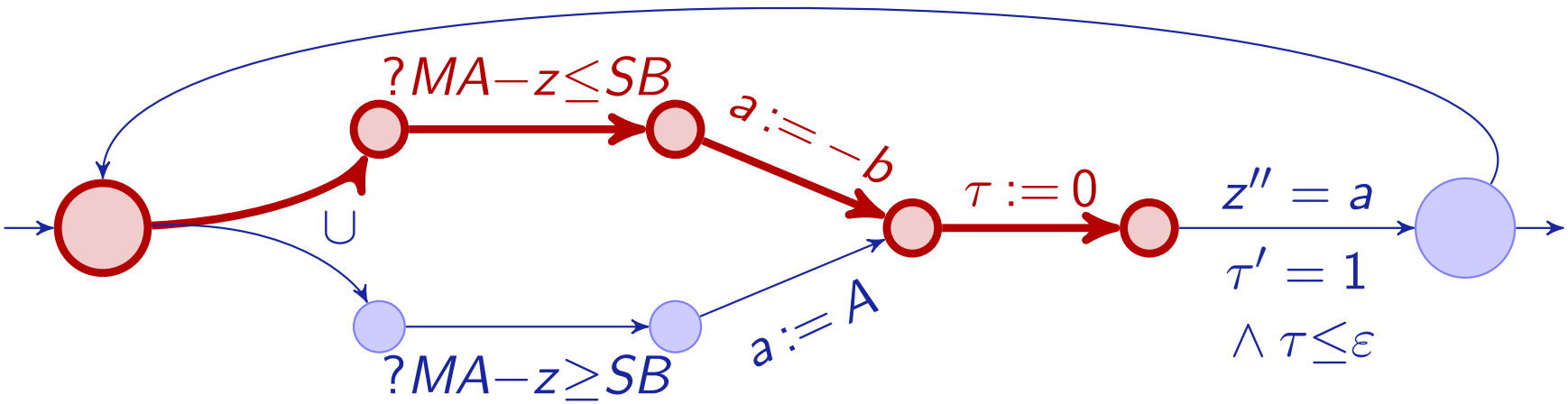
Branching Transitions in Hybrid Programs: ETCS

$$
\begin{aligned}
\text{system} &\equiv (\mathit{cor}\,;\mathit{drive})^* \\
\mathit{cor} &\equiv (?MA - z \leq SB; a := -b) \cup (?MA - z \geq SB; a := A) \\
\mathit{drive} &\equiv \tau := 0; (z' = v, v' = a, \tau' = 1 \wedge v \geq 0 \wedge \tau \leq \varepsilon)
\end{aligned}
$$



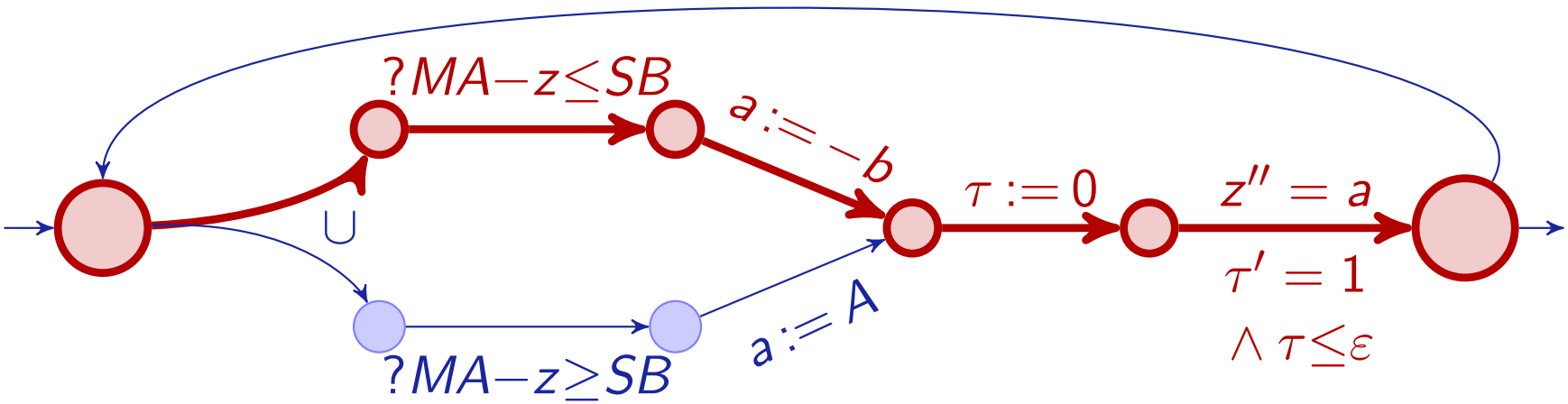
Branching Transitions in Hybrid Programs: ETCS

$$\begin{aligned} \text{system} &\equiv (cor\,;\, drive)^* \\ cor &\equiv (?MA - z \leq SB; a := -b) \cup (?MA - z \geq SB; a := A) \\ drive &\equiv \tau := 0; (z' = v, v' = a, \tau' = 1 \wedge v \geq 0 \wedge \tau \leq \varepsilon) \end{aligned}$$



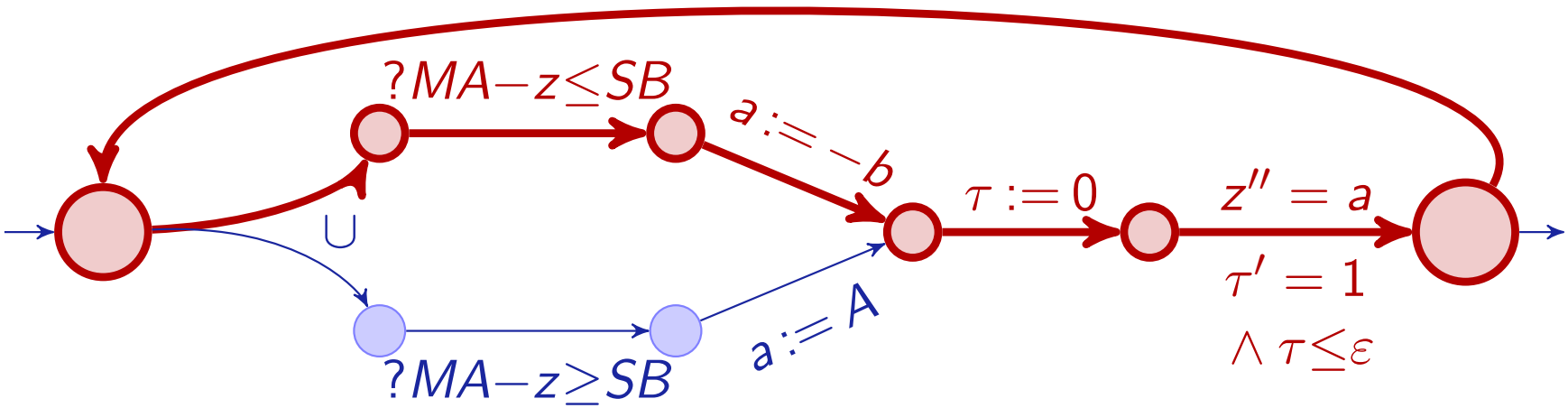
Branching Transitions in Hybrid Programs: ETCS

$$\begin{aligned} \text{system} &\equiv (cor\,;\, drive)^* \\ cor &\equiv (?MA - z \leq SB; a := -b) \cup (?MA - z \geq SB; a := A) \\ drive &\equiv \tau := 0; (z' = v, v' = a, \tau' = 1 \wedge v \geq 0 \wedge \tau \leq \varepsilon) \end{aligned}$$



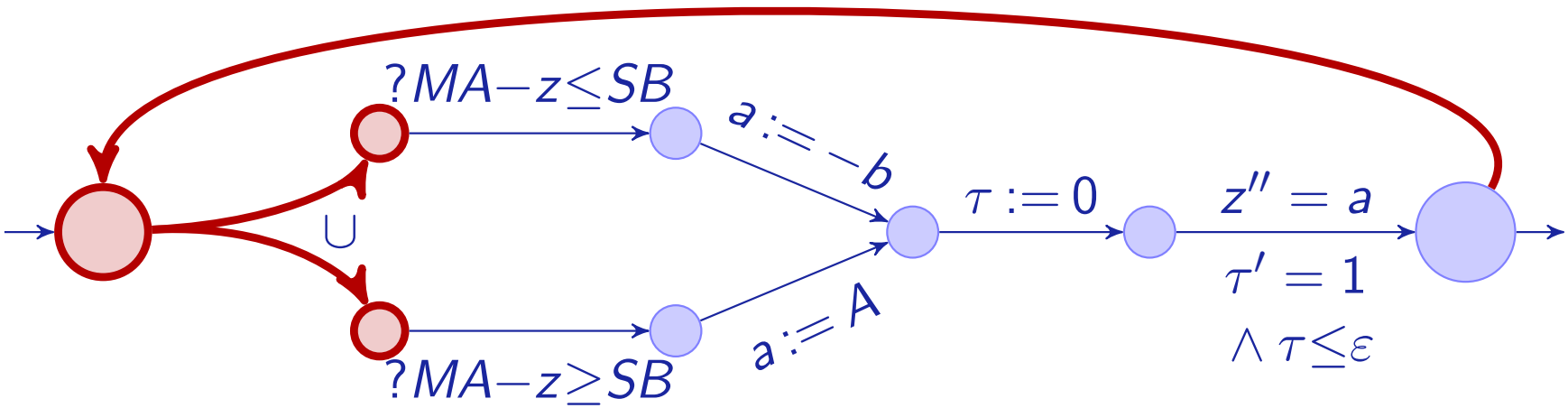
Branching Transitions in Hybrid Programs: ETCS

$$
\begin{aligned}
\text{system} &\equiv (cor\,;drive)^* \\
cor &\equiv (?MA - z \leq SB; a := -b) \cup (?MA - z \geq SB; a := A) \\
drive &\equiv \tau := 0; (z' = v, v' = a, \tau' = 1 \wedge v \geq 0 \wedge \tau \leq \varepsilon)
\end{aligned}
$$



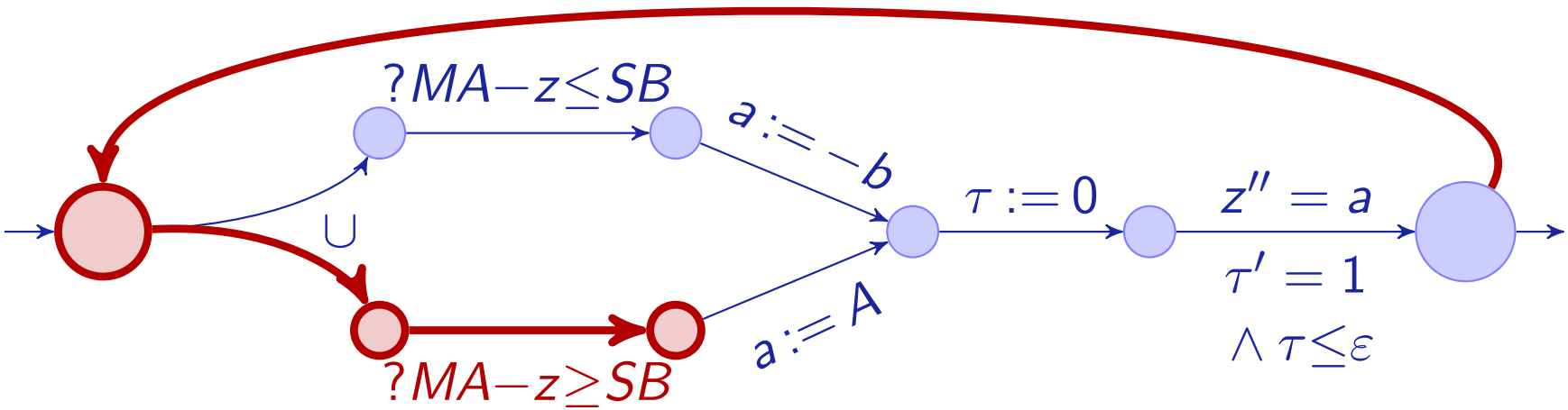
Branching Transitions in Hybrid Programs: ETCS

$$\begin{aligned} \text{system} &\equiv (\mathit{cor}\,;\mathit{drive})^* \\ \mathit{cor} &\equiv (?MA - z \leq SB; a := -b) \cup (?MA - z \geq SB; a := A) \\ \mathit{drive} &\equiv \tau := 0; (z' = v, v' = a, \tau' = 1 \wedge v \geq 0 \wedge \tau \leq \varepsilon) \end{aligned}$$



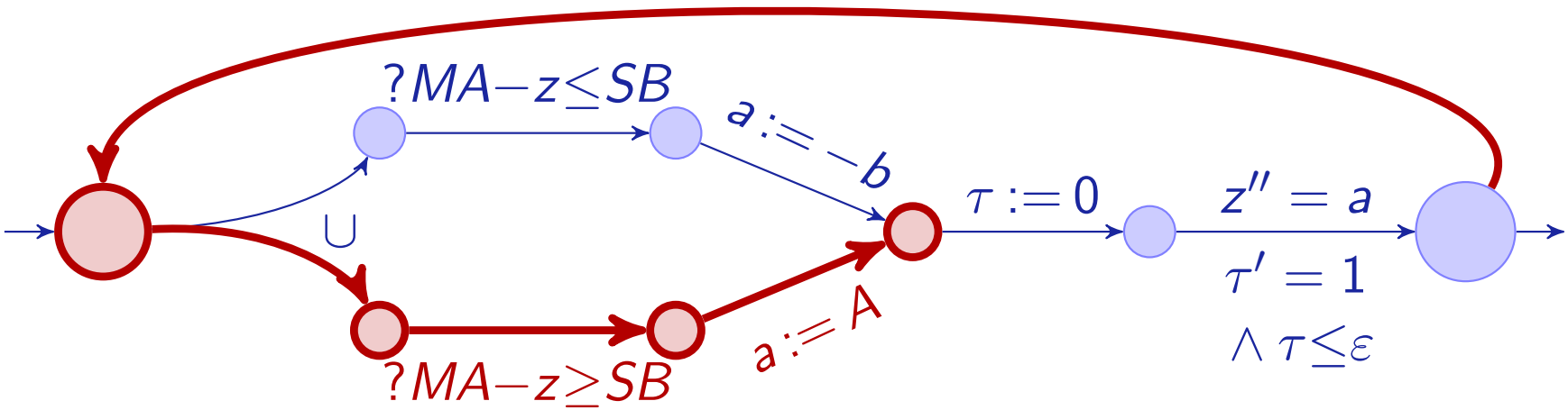
Branching Transitions in Hybrid Programs: ETCS

$$\begin{aligned} \text{system} &\equiv (\mathit{cor}\,;\mathit{drive})^* \\ \mathit{cor} &\equiv (?MA - z \leq SB; a := -b) \cup (?MA - z \geq SB; a := A) \\ \mathit{drive} &\equiv \tau := 0; (z' = v, v' = a, \tau' = 1 \wedge v \geq 0 \wedge \tau \leq \varepsilon) \end{aligned}$$



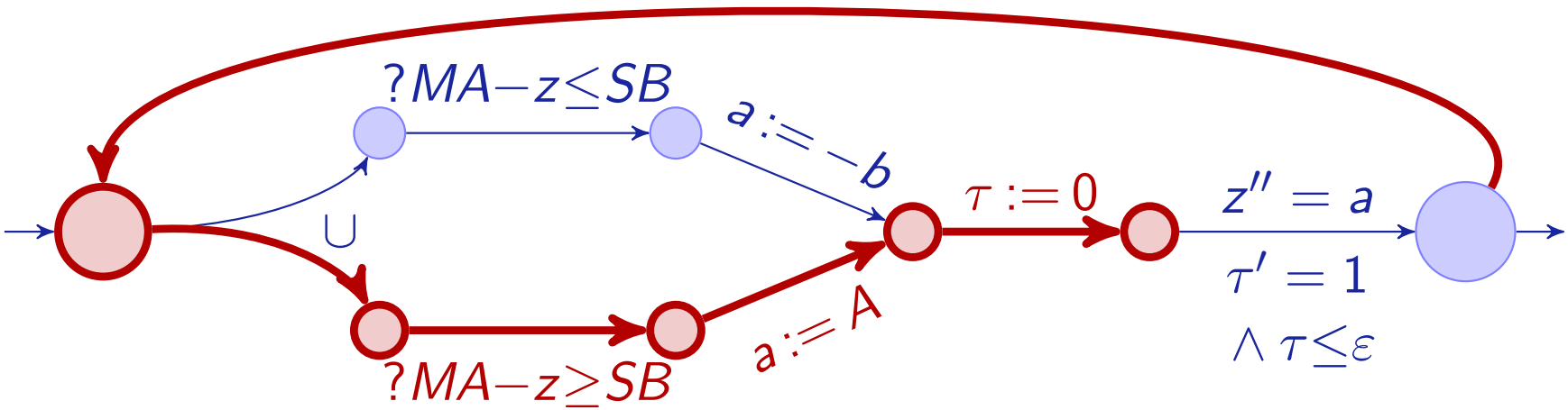
Branching Transitions in Hybrid Programs: ETCS

$$
\begin{aligned}
\text{system} &\equiv (\mathit{cor}\,;\mathit{drive})^* \\
\mathit{cor} &\equiv (?MA - z \leq SB; a := -b) \cup (?MA - z \geq SB; a := A) \\
\mathit{drive} &\equiv \tau := 0; (z' = v, v' = a, \tau' = 1 \wedge v \geq 0 \wedge \tau \leq \varepsilon)
\end{aligned}
$$



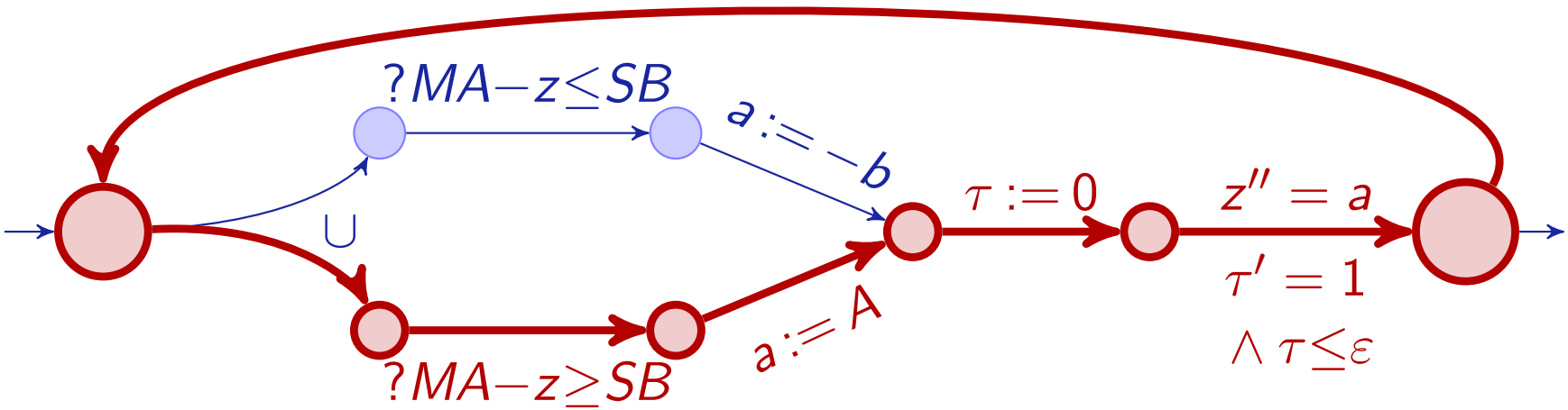
Branching Transitions in Hybrid Programs: ETCS

$$\begin{aligned} \text{system} &\equiv (\mathit{cor}\,;\mathit{drive})^* \\ \mathit{cor} &\equiv (?MA - z \leq SB; a := -b) \cup (?MA - z \geq SB; a := A) \\ \mathit{drive} &\equiv \tau := 0; (z' = v, v' = a, \tau' = 1 \wedge v \geq 0 \wedge \tau \leq \varepsilon) \end{aligned}$$



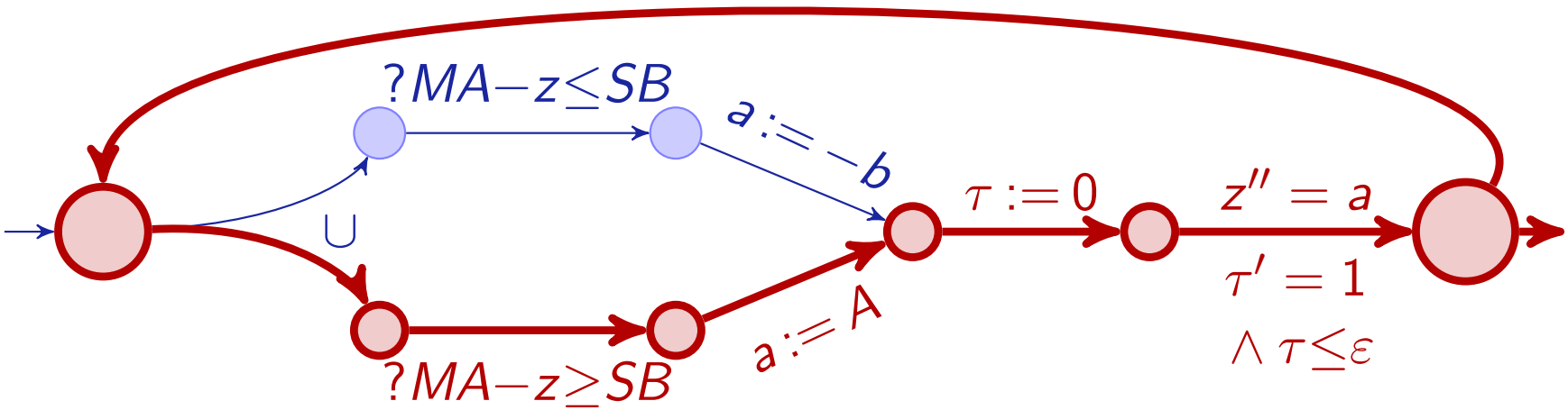
Branching Transitions in Hybrid Programs: ETCS

$$\begin{aligned} \text{system} &\equiv (cor\,;\, drive)^* \\ cor &\equiv (?MA - z \leq SB; a := -b) \cup (?MA - z \geq SB; a := A) \\ drive &\equiv \tau := 0; (z' = v, v' = a, \tau' = 1 \wedge v \geq 0 \wedge \tau \leq \varepsilon) \end{aligned}$$



Branching Transitions in Hybrid Programs: ETCS

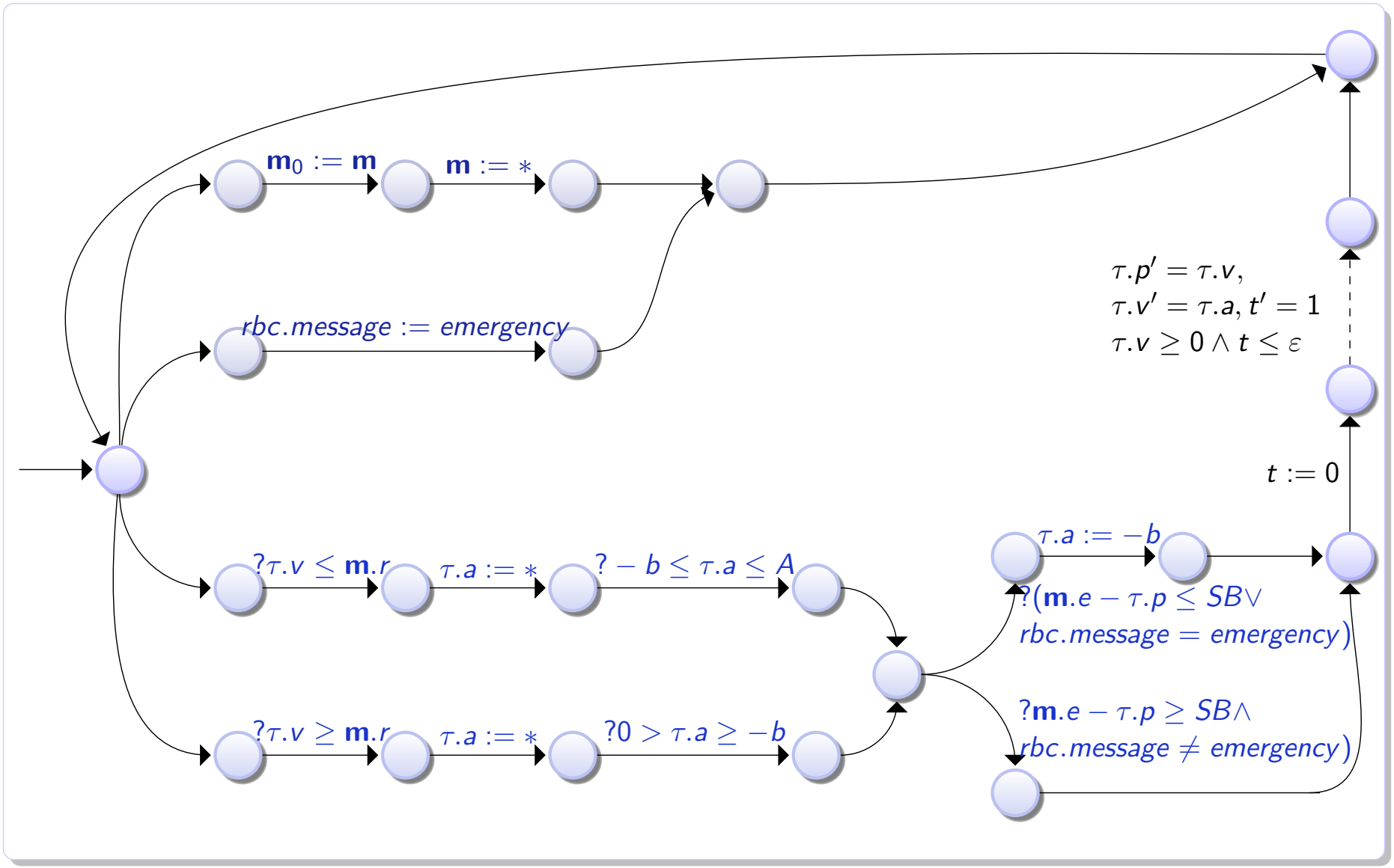
$$
\begin{aligned}
\text{system} &\equiv (cor\,;drive)^* \\
cor &\equiv (?MA - z \leq SB; a := -b) \cup (?MA - z \geq SB; a := A) \\
drive &\equiv \tau := 0; (z' = v, v' = a, \tau' = 1 \wedge v \geq 0 \wedge \tau \leq \varepsilon)
\end{aligned}
$$



ETCS Control

$$
\begin{array}{lll}
\text{ETCS:} & (\text{train} \cup \text{rbc})^* & \\
\text{train} & : & \text{spd};\ \text{atp};\ \text{move} \\
\text{spd} & : & (?\tau.v \leq \mathbf{m}.r;\ \tau.a := *;\ ? - b \leq \tau.a \leq A) \\
 & & \cup (?\tau.v \geq \mathbf{m}.r;\ \tau.a := *;\ ?0 > \tau.a \geq -b) \\
\text{atp} & : & SB := \frac{\tau.v^2 - \mathbf{m}.d^2}{2b} + \left(\frac{A}{b} + 1\right)\left(\frac{A}{2}\varepsilon^2 + \varepsilon\,\tau.v\right); \\
 & & (?(\mathbf{m}.e - \tau.p \leq SB \vee rbc.message = emergency);\ \tau.a := -b) \\
 & & \cup (?\mathbf{m}.e - \tau.p \geq SB \wedge rbc.message \neq emergency) \\
\text{move} & : & t := 0;\ (\tau.p' = \tau.v, \tau.v' = \tau.a, t' = 1 \wedge \tau.v \geq 0 \wedge t \leq \varepsilon) \\
\text{rbc} & : & (rbc.message := emergency) \\
 & & \cup \big(\mathbf{m}_0 := \mathbf{m}; \mathbf{m} := *; \\
 & & \quad ?\mathbf{m}.r \geq 0 \wedge \mathbf{m}.d \geq 0 \wedge \mathbf{m}_0.d^2 - \mathbf{m}.d^2 \leq 2b(\mathbf{m}.e - \mathbf{m}_0.e)\big)
\end{array}
$$

ETCS Control Transition Structure




A. Platzer.
Differential dynamic logic for verifying parametric hybrid systems.
In N. Olivetti, editor, *TABLEAUX*, volume 4548 of *LNCS*, pages 216–232. Springer, 2007.


A. Platzer.
Differential dynamic logic for hybrid systems.
J. Autom. Reas., 41(2):143–189, 2008.


A. Platzer.
Logical Analysis of Hybrid Systems: Proving Theorems for Complex Dynamics.
Springer, Heidelberg, 2010.