Master Comasic
École Polytechnique

Stage de Recherche à
Carnegie Mellon University

Rapport Non Confidentiel

Delay Differential Logic for Hybrid Systems with Delay

Lorenz Sahlmann

Mars – Août 2016

Tuteur de stage
Prof. Dr. André Platzer
Carnegie Mellon University

Enseignant référent
Prof. Eric Goubault
École Polytechnique

Computer Science Department
Carnegie Mellon University
Pittsburgh, PA 15213-3891
USA

Abstract

Cyber physical systems incorporate the connection between the physical world and computing devices. This connection (e.g. given by a computer network) often needs to be considered in the system model, if it introduces a delay with non-negligible effects.

In this work we provide a program notation for hybrid systems with delay and introduce a logic to reason about such systems, called *delay differential dynamic logic* ($\mathsf{dd}\mathcal{L}$). It extends *differential dynamic logic* ($\mathsf{d}\mathcal{L}$) with syntax, semantics and axiomatization for dealing with delay differential equations.

We prove the soundness of the axiomatization and give examples demonstrating its usefulness.

Résumé

Le système cyber-physique est le trait d'union entre le monde physique et le périphérique numérique. Cette connexion (par ex. donnée par un réseau numérique) doit souvent être prise en considération dans le modèle du système, si ce dernier introduit un retard avec des effets non négligeables.

Dans cet oeuvre, une notation de programme sera donnée pour des systèmes hybrides avec un retard, ainsi qu'une logique de raisonnement pour de tels systèmes, appelée *logique dynamique différentielle retardée* ($\mathsf{dd}\mathcal{L}$), sera introduite. Elle étend la *logique dynamique différentielle* ($\mathsf{d}\mathcal{L}$) avec une syntaxe, une sémantique et une axiomatisation pour gérer les équations différentielles retardées.

La preuve de correction de l'axiomatisation est également présentée et quelques exemples sont avancés pour montrer son utilité.

Déclaration d'intégrité relative au plagiat

Je soussigné SAHLMANN, Lorenz certifie sur l'honneur :

1. Que les résultats décrits dans ce rapport sont l'aboutissement de mon travail.
2. Que je suis l'auteur de ce rapport.
3. Que je n'ai pas utilisé des sources ou résultats tiers sans clairement les citer et les référencer selon les régles bibliographiques préconisées.

Mention à recopier : Je déclare que ce travail ne peut être suspecté de plagiat.

Date : 18 septembre 2016

Signature :

Acknowledgement

I would like to express my sincere thanks to Prof. Dr. André Platzer for supervising this work, the inspirational discussions and his very helpful suggestions and comments.

Also, I would like to thank him and the Carnegie-Mellon-University for having accepted me as a research intern.

My thanks go to my professor Eric Goubault, who introduced me to delay differential equations and who presented me to Prof. Platzer.

A grateful word of thanks to the *Chair Thales* and *La Fondation de l'Ecole Polytechnique* for supporting me financially throughout the period of the internship.

Contents

1 Introduction

Dynamical systems are a mathematical model which describe the evolution of a system's state over time.

There are *discrete dynamical systems*, obeying difference equations and discrete state transition relations, and *continuous dynamical systems*, whose states evolve continuously, described by a (ordinary) differential equation.

A fusion of both are *hybrid (dynamical) systems*, which combine discrete and continuous dynamics with conditional switching, nondeterminism and repetition.

The fashionable notion of *cyber-physical systems* (CPS) is broadly used to describe technical systems which apply the discrete dynamics of digital computation (the cyber part) to control the continuous dynamics of physical processes.

Hybrid systems are well suited for capturing the complex behavior of CPS in a mathematical model.

With the ongoing emergence of technology, cyber-physical systems are getting perceptibly more present in our surroundings and begin to largely interfere in our everyday life. While the complexity of these systems is growing, more and more control and responsibility is handed off to such technical systems. Thus the question of their "safety" is getting increasingly important.

This demands the definition and specification of what is a reasonable and appropriate behavior for a system, including not only classical safety properties (something never happens), but also liveness (something eventually happens), controllability and reactivity.

The task of *verification* is to show that the system satisfies its specification and to establish guarantees for its safety- and performance-critical correctness.

However, the complexity of dynamical system usually leads to a (uncountably) infinite state space, what means that any finite number of tests cannot cover all possible states reachable by execution of the system. Thus a finite number of tests cannot prove safety!

Formal methods provide a means to systematically obtain proofs of specifications, if the system can be described as a model in a certain formal language. This way, safety certificates can be established.

Classical examples for CPS include all applications of automatic control, such as in robotics, automotive (self-driving cars), aviation, railway or power plants, but also models of electrical circuits, chemical and biological processes, as well as medical models. All these examples have in common that they admit events which can be seen as discrete in relation to their continuous evolution part.

A language for the specification and verification of correctness (such as safety and liveness) properties for hybrid systems is *differential dynamic logic* ($\mathsf{d}\mathcal{L}$) (see [Pla12a] for a concise overview).

It provides syntax and semantics for hybrid programs and logic formulas as well as a proof calculus to formally reason about hybrid systems. This logic is based on first-order modal logic and dynamic logic and relies on first-order real arithmetic. The tutorial [Que+16] shows some examples of systems modeled and proved in $\mathsf{d}\mathcal{L}$.

With differential forms, $\mathsf{d}\mathcal{L}$ provides a powerful tool to draw conclusions about ordinary differential equations by differential invariants (an induction principle for differential equations), differential substitutions and differential ghosts.

Some important theoretical results, such as *soundness* (everything provable is true) and *completeness* (everything true is provable) have been established for differential dynamic logic.

Another important property of $\mathsf{d}\mathcal{L}$ is its compositionality. Its denotational semantics (of models and formulas) are defined as functions of their components. This allows a structural decomposition of proofs by splitting complex systems in their parts. The completeness property assures that this decomposition is always possible and successful. Moreover, this modularity makes $\mathsf{d}\mathcal{L}$ extendable. New proof rules can be added to the proof calculus, to improve its deductive power.

Different formulations of $\mathsf{d}\mathcal{L}$ have been presented. The earliest, given in [Pla10b], is a sequent calculus (in Gentzen style), which is tuned for automatic proof search. It is implemented in the proof assistant KeYmaera.

The later, Hilbert-type axiomatic formulation of $\mathsf{d}\mathcal{L}$ (cf. [Pla15a; Pla12b; Pla12a]) is based on uniform substitution and bound variable renaming, which allows a much more concise programmatic implementation. This is done in form of the interactive theorem prover KeYmaera X.

Moreover, some extensions to $\mathsf{d}\mathcal{L}$ have been presented. *Differential-algebraic dynamic logic* (DAL) adds differential-algebraic equations and constraints to $\mathsf{d}\mathcal{L}$ [Pla10a], whereas *differential temporal dynamic logic* (dTL) is based on a trace semantics, which allows to specify temporal properties of a hybrid system [Pla07]. *Stochastic differential dynamic logic* (S$\mathsf{d}\mathcal{L}$) deals with stochastic hybrid systems, which add stochastic differential equations to hybrid systems [Pla11]. Distributed hybrid systems can be verified using *quantified differential dynamic logic* (Q$\mathsf{d}\mathcal{L}$), cf. [Pla10c]. For *differential game logic* ($\mathsf{dG}\mathcal{L}$) see [Pla15b].

Differential dynamic logic and its extensions have successfully demonstrated their usability by application to a number of real-world problems. Examples include obstacle avoidance in robotics [Mit+16] and the design of a safe controller for medical surgery robots [Kou+13]. Contributions to the important field of self-driving cars have been made by the verification of cruise controllers [LPN11; Loo+13], controllers for intersections [LP11] and speed limit [MLP12].

Safety is paramount in aviation [Gho+] and KeYmaera was used to verify aircraft collision-avoidance systems [Jea+15b; Jea+15a; LRP13]. For the European Train Control System (ETCS), controllability, safety, liveness, and reactivity properties have been proved [PQ09].

Models in $\mathsf{d}\mathcal{L}$ are limited to ordinary differential equations. In this report, we will study a more general class of dynamical systems, called *time-delay systems* (TDS) [Ric03], which extend dynamical systems by obeying *delay differential equations* (DDEs). Delay differential equations (also called differential-difference equations) belong to the broader class of *functional differential equations* (FDEs).

Delay is mainly an applied problem. Most real world CPS incorporate sensors and actuators in a feedback loop. This system-intern communication introduces a delay, for example due to computation time, sensing sample-intervals or network transfer, whose effects can often not be ignored.

Thus, time-delay systems have become of high interest in research and application in the recent decades, especially in the systems and control community. By taking the delay into account, DDEs allow to formulate more realistic models with better performance.

Prototypical examples of time-delay systems include networked control systems (NCS) and tele-operated systems, general communication networks, robotics, combustion engines and manufacturing processes. See [GN03] for a number of examples.

The theory of DDEs is more complex than for ODEs. As functional equations, delay differential equations have an infinite dimensional state space. This is due to the fact that they need to remember (a part of) their own past to define their current state.

Even small delays can have a huge impact on the system's behavior (cf. Example 3.12), in particular on its stability, which is an important property for many applications. The presence of a delay can both be destabilizing and stabilizing.

To limit the complexity, one may restrict to point-wise delays, what leads to the special class of delay differential equations with multiple, constant delay. In this work, we will restrict to this type of DDE and present a logic to formally reason about time-delay systems.

2 Introduction to Logic

Logics is a framework for studying *"rules of argument"* [Hod01] and can be used as a means for the specification of properties of hybrid programs.

We give a concise overview about the concepts of logics on which the later chapters are based and which are essential for their understanding.

2.1 Propositional Logic

The most familiar concept of logics is *propositional logic* (eg. [HR04]). It deals with *declarative sentences*, so-called *propositions*, which can be declared as either *true* or *false*, their truth value. Declarative sentences can be a composition of atomic (indecomposable) sentences, using the logical connectives: negation (¬), conjunction (∧), disjunction (∨), implication (→) or equivalence (↔).

Each connection admits a new truth value depending on the the truth values of its sub-sentences. They are defined by truth tables (see Figure 2.1), specifying the result for each combination of input values. These operators can be partly expressed by each other, demanding, for example, only the definition of ¬, ∧, ∨.

2.2 First-Order Logic

First-order logic (FOL) extends propositional logic with quantifiers, variables, functions and predicates. We introduce first-order logic similar to the definitions given in [Pla10b] and [HR04]. More subtleties are adapted from [Hod01], [Bus98] and [Rau10].

FOL is a class of logics, whose incarnations differ in their set of symbols and their roles, but follow all the same basic structure. Besides the logical connectives from propositional logic, first-order logic comprises extralogical symbols (functions and predicates) as well as logical variables, which altogether form its alphabet.

¬	*false*	*true*
	true	*false*

∧	*false*	*true*
false	*false*	*false*
true	*false*	*true*

∨	*false*	*true*
false	*false*	*true*
true	*true*	*true*

→	*false*	*true*
false	*true*	*true*
true	*false*	*true*

↔	*false*	*true*
false	*true*	*false*
true	*false*	*true*

Figure 2.1: Definition of logical connectives.

Logical variables provide a means of abstraction and refer to objects about which one wants to reason. They are usually called x, y, z.

Functions take values as argument and give back a new value, which can be of any type. Function symbols stand for a function and are denoted by f, g, h.

In contrast to functions, *predicates* return a truth value (*true* or *false*), depending on the values of their arguments. Predicate symbols are usually written as p, q, r.

The *arity* of a function or predicate symbol is the number of arguments it expects. Functions can be of arity zero, i.e. without any argument. For clarity, we will treat these so-called *constants* (or nullary functions) separately and write their symbols as a, b, c.

2.2.1 Syntax

First-order logic inductively defines a syntax of well-formed *logical formulas* over an alphabet containing different kinds of symbols.

The syntax only defines rules which specify how the given symbols can be textually combined. At this stage, they have no specific meaning yet.

The alphabet $\Sigma \cup V$ comprises the *signature* Σ of the theory, which is a set of predicate, function and constant symbols, and the set of logical variable symbols V.

Terms are the expressions which denote feasible arguments for functions and predicates.

Definition 2.1 (Terms)**.** For a given *signature* Σ and set of variable symbols V, the well-formed terms are either logical variables, constants or functions applied to terms: The set of all *terms* $\mathrm{Trm}(\Sigma, V)$ is the smallest set with

1. If $x \in V$ then $x \in \mathrm{Trm}(\Sigma, V)$.
2. If $c \in \Sigma$ is a constant symbol, then $c \in \mathrm{Trm}(\Sigma, V)$.
3. If $f \in \Sigma$ is a function symbol of arity $n \geq 1$ and $\theta_i \in \mathrm{Trm}(\Sigma, V)$ for $i = 1, \ldots, n$, then $f(\theta_1, \ldots, \theta_n) \in \mathrm{Trm}(\Sigma, V)$.

This definition can alternatively be written as grammar in Backus-Naur form

$$\theta, \theta_i ::= x \mid c \mid f(\theta_1, \ldots, \theta_n)$$

where x is ranging over the set of logical variables V, c over the nullary functions in Σ and f over the elements in Σ with arity $n \geq 1$.

Formulas are expressions which admit a truth value.

Definition 2.2 (First-Order Formulas)**.** For given Σ and V, the well-formed *formulas* of a first-order logic are words formed by recursive combination of signature-symbols with logical operator symbols: The set of formulas $\mathrm{Fml}_{\mathrm{FOL}}(\Sigma, V)$ is the smallest set with

1. If $p \in \Sigma$ is a predicate symbol of arity $n \geq 0$ and $\theta_i \in \mathrm{Trm}(\Sigma, V)$ for $i = 1, \ldots, n$, then $p(\theta_1, \ldots, \theta_n) \in \mathrm{Fml}_{\mathrm{FOL}}(\Sigma, V)$.

2. If $\phi, \psi \in \mathrm{Fml}_{\mathrm{FOL}}(\Sigma, V)$, then $\neg\phi, (\phi \wedge \psi), (\phi \vee \psi), (\phi \to \psi) \in \mathrm{Fml}_{\mathrm{FOL}}(\Sigma, V)$.
3. If $\phi \in \mathrm{Fml}_{\mathrm{FOL}}(\Sigma, V)$ and $x \in V$ then $(\forall x\, \phi) \in \mathrm{Fml}_{\mathrm{FOL}}(\Sigma, V)$ and $(\exists x\, \phi) \in \mathrm{Fml}_{\mathrm{FOL}}(\Sigma, V)$.

We may again write this as

$$\phi, \psi ::= p(\theta_1, \dots, \theta_n) \mid \neg\phi \mid \phi \wedge \psi \mid \phi \vee \psi \mid \phi \to \psi \mid \forall x\, \phi \mid \exists x\, \phi \tag{2.1}$$

where $p \in \Sigma$ ranges over the predicate symbols (of arity $k \geq 1$), $\theta_i \in \mathrm{Trm}(\Sigma, V)$ over the terms and $x \in V$ over the variable symbols.

2.2.2 Semantics

The *semantics* of a first-order logic specifies the meaning of each symbol occurring in terms and formulas with the goal to assign a *truth value* to a full formula. Moreover, this requires the *assignment* of a concrete value to each variable symbol, chosen from a *universe*, the domain of discourse containing all objects of consideration.

Definition 2.3 (Universe)**.** A non-empty set D containing the concrete objects of discourse, is called *universe*.

Definition 2.4 (Interpretation)**.** An *interpretation* I assigns concrete elements to their corresponding symbols in a given signature Σ. It consists of:

1. for each constant symbol $c \in \Sigma$ a value $I(c) \in D$,
2. for each function symbol $f \in \Sigma$ of arity $k \geq 1$ a function $I(f)\colon D^k \to D$ with k arguments,
3. for each predicate symbol p of arity $k \geq 1$ a relation $I(p) \subseteq D^k$.

A predicate p is *true* at position $(d_1, \dots, d_n) \in D^k$ if and only if $(d_1, \dots, d_n) \in I(p)$. An alternative but equivalent formulation uses the characteristic function of the relation $I(p)\colon D^n \to \{\mathit{true}, \mathit{false}\}$, where p is true at $(d_1, \dots, d_n) \in D^n$ if and only if $I(p)(d_1, \dots, d_n) = \mathit{true}$.

As mentioned above, logical variables are only place holders and need, at some stage, refer to a concrete value. Their interpretation is realized by an assignment.

Definition 2.5 (Assignment)**.** An *assignment* is a map $\nu\colon V \to D$ which assigns a value of the universe to each variable symbol $x \in V$.

Definition 2.6 (Model)**.** A *model* is the triple $M = (D, I, \nu)$ of a universe D, an interpretation I and an assignment ν.

We write $M[x \mapsto a]$ for the model which differs from the model M only in the assignment to the logical variable x, to which the value $a \in D$ is allocated.

Given the information of a model, we can finally evaluate a formula, i.e. decide on its truth value.

The valuation of terms and formulas is again defined inductively.

Definition 2.7 (Valuation of Terms)**.** For $\mathrm{Trm}(\Sigma, V)$ a *valuation* $[\![\phi]\!]_\nu^I$ is defined by

1. $[\![x]\!]_\nu^I = \nu(x)$ for each logical variable $x \in V$
2. $[\![c]\!]_\nu^I = I(c)$ for each constant $c \in \Sigma$
3. $[\![f(\theta_1, \ldots, \theta_n)]\!]_\nu^I = I(f)([\![\theta_1]\!]_\nu^I, \ldots, [\![\theta_n]\!]_\nu^I)$ for each function symbol $f \in \Sigma$ of arity $n \geq 1$.

The meaning of the logical connective symbols, how they preserve a truth value, is defined in the truth tables, given in Figure 2.1.

Definition 2.8 (Valuation of First-Order Formulas)**.** The *valuation* of the first-order formulas $\mathrm{Fml}_{\mathrm{FOL}}(\Sigma, V)$ in the interpretation I under the assignment ν is given by

1. $[\![p(\theta_1, \ldots, \theta_n)]\!]_\nu^I = I(p)([\![\theta_1]\!]_\nu^I, \ldots, [\![\theta_n]\!]_\nu^I)$ for predicates p
2. the conjunction $[\![\theta \wedge \eta]\!]_\nu^I = \mathit{true}$ iff $[\![\theta]\!]_\nu^I = \mathit{true}$ and $[\![\eta]\!]_\nu^I = \mathit{true}$.
3. the disjunction $[\![\theta \vee \eta]\!]_\nu^I = \mathit{true}$ iff $[\![\theta]\!]_\nu^I = \mathit{true}$ or $[\![\eta]\!]_\nu^I = \mathit{true}$.
4. the negation $[\![\neg\theta]\!]_\nu^I = \mathit{true}$ iff $[\![\theta]\!]_\nu^I \neq \mathit{true}$.
5. the implication $[\![\theta \to \eta]\!]_\nu^I = \mathit{true}$ iff $[\![\theta]\!]_\nu^I \neq \mathit{true}$ or $[\![\eta]\!]_\nu^I = \mathit{true}$.
6. the universal quantifier $[\![\forall x\, \theta]\!]_\nu^I = \mathit{true}$ iff $[\![\theta]\!]_\nu^I = \mathit{true}$ for all $a \in \mathbb{R}$.
7. the existential quantifier $[\![\exists x\, \theta]\!]_\nu^I = \mathit{true}$ iff $[\![\theta]\!]_\nu^I = \mathit{true}$ for some $a \in \mathbb{R}$.

The statement $[\![\theta]\!]_\nu^I = \mathit{true}$ can also be written as $I, \nu \models \theta$, the so-called *satisfaction relation*. One says that (I, ν) *satisfies* ϕ, that (I, ν) is a model for ϕ or that ϕ is *true* in I under ν.

A formula ϕ is called *valid* (or a *tautology*) if $I, \nu \models \phi$ for every model M (i.e. for all possible interpretations I and assignments ν). In this case, one writes shortly $\models \phi$.

2.2.3 Variable Binding and Substitution

Quantifiers in formulas bind occurrences of variables in such way that a *bound variable* does not need an assignment in order to determine a truth value for the formula. In contrast, a *free variable* requires an assignment.

Definition 2.9 (Variable binding)**.** A variable x in a formula ϕ, which occurs in subformulas of the form $\forall x\, \psi$ or $\exists x\, \psi$, is *bound* in ϕ. If it appears in ϕ outside the scope (ψ) of a quantifier, it is called *free.*

Variables are place holders and hence demand a means for being replaced with concrete information. With a little restriction, it is possible to substitute free variables by another term.

Definition 2.10 (Admissible substitution)**.** The term θ is an *admissible substitution* for x in $\phi(x)$, if no free occurrence of x in ϕ is in the scope of a $\forall y$ or $\exists y$ for any variable y appearing in t.

In this case, we define the *substitution* $\phi(\theta/x)$ of the variable x by the admissible term θ in the formula ϕ as the formula obtain from ϕ by replacing each occurrence of the free variable x outside the scope of a quantifier by θ.

This restriction is necessary for the soundness of the quantifier axioms given in the next section.

We agree on the following notational convention (which is not to be confused with the syntax for functions and predicates with arguments): When me mention $\phi(x)$ and $\phi(\theta)$ in the same context, we mean that x is a free variable of the formula ϕ and that $\phi(\theta)$ is $\phi(\theta/x)$.

2.2.4 Proof Theory

Given an interpretation and assignment, the semantics of a logic allow to compute the truth value of a formula. This means, checking satisfaction is easy. However to check the validity of a formula by the semantics is difficult. We would need to check all possible interpretations and assignments.

To tackle that problem, one introduces the concept of a proof calculus, which allows to show the validity of a formula. A *formal proof* derives an obviously valid formula from the initial formula, using a set of *sound* derivation rules.

In general, it can be difficult to find a proof. But once a proof is found, it can easily be verified by the validity of the proof rules. This means, as opposed to the semantics, showing that a formula is not valid using the proof calculus, is difficult. Using the semantics, it is sufficient to find a counter-example. But how can one show that there is no proof? A counterexample is a witness for non-validity, a (formal) proof is a witness for validity.

These two characterizations of truth, semantics and proof calculus, need to be equivalent. A proof calculus is *sound* if invalid sequents are not derivable and *complete* if every valid sequent can be derived.

There are different styles of formal logical argumentation. We consider so-called Hilbert deduction systems, in which a proof can be represented.

Hilbert Calculus

A Hilbert-style deduction system is based on a finite set of selected tautologies with the role of axioms and only a small number of inference rules.

In such a calculus, a formal *proof* is a derivation of the formula to be shown (conclusion) from the axioms and premises by applying in each step one of the deduction rules. In such a proof, every line is again an unconditional tautology. A formula ϕ is called *derivable* from the premises in the given calculus, if there exists a derivation with conclusion ϕ.

$$\begin{array}{lll}
\Lambda 1 & \phi \to (\psi \to \phi) & \\
\Lambda 2 & \big(\phi \to (\psi \to \varphi)\big) \to \big((\phi \to \psi) \to (\phi \to \varphi)\big) & \\
\Lambda 3 & (\neg\phi \to \neg\psi) \to \big((\neg\phi \to \psi) \to \phi\big) & \\
\Lambda 4 & \forall x\,\phi(x) \to \phi(y) & (y \text{ admissible for substitution}) \\
\Lambda 5 & \forall x\,(\phi \to \psi) \to (\phi \to \forall x\,\psi) & (x \text{ is not free in } \phi)
\end{array}$$

$$\text{MP} \quad \frac{\phi \to \psi \quad \phi}{\psi} \qquad\qquad \forall \quad \frac{\phi}{\forall x\,\phi}$$

Figure 2.2: A complete axiomatization of FOL and Hilbert-calculus deduction rules.

The axiomatization consists of logical *axiom schemes.* These describe the countably infinite number of axioms the deduction system comprises. A concrete axiom is obtained by replacing the placeholders by concrete formulas.

There are many possible choices for an axiom system. However the selection should be sufficiently strong, such that each tautology is derivable by the given inference rules.

A possible sound and complete axiomatization for first-order logic is given in [Bim14] and shown in Figure 2.2. It combines axioms for classical propositional logic with axioms for quantifiers and relies only on two deduction rules: *modus ponens* (MP), which infers formula ψ from the hypotheses ϕ, and $\phi \to \psi$ and *quantifier-generalization* ($\forall$).

2.3 Modal Logic

In propositional and first-order logic, each formula is either *true* or *false* for a given model. In this section, we present another concept of logics, called *modal logic*, which allows different *modes of truth.* An example is truth with respect to time. To that end, modal logic extends the language of propositional logic with two new unary connectives, $\Box$ and $\Diamond$, whose exact meaning depends on a specific incarnation of modal logics. We define modal logic as in [HR04; JJ98].

Definition 2.11 (Syntax)**.** The syntax of *modal logic* is defined by the grammar

$$\phi, \psi ::= \mathit{false} \mid \mathit{true} \mid p \mid \neg\phi \mid \phi \wedge \psi \mid \phi \vee \psi \mid \phi \to \psi \mid \phi \leftrightarrow \psi \mid \Box\phi \mid \Diamond\phi$$

with an atomic formula p from the set of propositions P.

Definition 2.12 ((Kripke) Model)**.** A *model* $M = (W, \rho, L)$ of modal logic consists of

1. a nonempty set of *worlds/states* W
2. an accessibility relation $\rho \subseteq W \times W$ between the worlds
3. a labeling function $L\colon W \to \mathcal{P}(P)$

We write $\rho(\nu, \omega)$ for $(\nu, \omega) \in \rho$, meaning that the world ω is accessible from ν.

Definition 2.13 (Semantics of modal formulas)**.** The semantics of modal formulas is defined by a *satisfaction relation.* It specifies inductively the truth value of each symbolic expression with regard to the worlds. A formula ϕ is *true* in a world ν or equivalently the world ν *satisfies* formula ϕ, iff $\nu \models \phi$.

1. $\nu \not\models \mathit{false}$
2. $\nu \models \mathit{true}$
3. $\nu \models p$ iff $p \in L(\nu)$
4. $\nu \models \neg\phi$ iff $\nu \not\models \phi$
5. $\nu \models \phi \wedge \psi$ iff $\nu \models \phi$ and $\nu \models \psi$
6. $\nu \models \phi \vee \psi$ iff $\nu \models \phi$ or $\nu \models \psi$
7. $\nu \models \phi \rightarrow \psi$ iff $\nu \models \psi$ whenever $\nu \models \phi$
8. $\nu \models \phi \leftrightarrow \psi$ iff $\nu \models \phi$ if and only if $\nu \models \psi$
9. $\nu \models \Box\phi$ iff $\omega \models \phi$ for each $\omega \in W$ with $\rho(\nu, \omega)$
10. $\nu \models \Diamond\phi$ iff there is a $\omega \in W$ with $\rho(\nu, \omega)$ such that $\omega \models \phi$

A model M satisfies a formula if and only if each world of the model satisfies the formula. In this case, we write $M \models \phi$ and say ϕ is *valid* in M. A formula ϕ is *valid*, if and only if it is true in every world of every model, written as $\models \phi$.

3 Delay Differential Equations

Differential equations are often used to describe the dynamics of a deterministic system, whose future behavior depends on the present state. For an *ordinary differential equation* (ODE), this state is an element of $\mathbb{R}^n$. The rate of change only depends on the current time instant.

In *delay differential equations* (DDEs) however, the system is influenced by the past through the appearance of a deviated time argument. This means that the current state needs to contain the previous evolution. This leads to a functional state space, whose elements are functions on a past time interval. For that reason, DDEs belong to the class of *functional differential equations* (FDEs).

DDEs often appear in automatic control, where a controller monitors the state of a system in order to make control decisions to adjust this state. If there is a delay between the observation and the control action, the differential equation describing the system not only depends on its current state, but also on its past. These previous values need to be specified in an initial condition, for at least the time of the longest delay.

Examples of phenomena which have been modeled using delay differential equations include epidemics, traffic flow and vibrations/chattering. See [Fal06] and the references therein for a number of examples.

Some methods to solve basic DDEs analytically are presented in [Fal06]. Numerical procedures are not as far developed as for ODEs. See [BZ13], or [Szc14] for a rigorous integration algorithm.

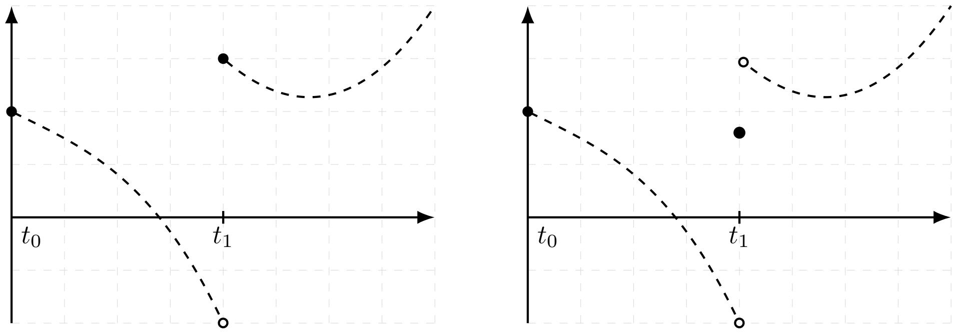
3.1 Piecewise Continuous Functions

The following definition is motivated by the character of evolution arising from hybrid systems. We define the main functional space of operation for the following chapters.

Definition 3.1 (Piecewise Continuously Differentiable)**.** Let $D = [a, b] \subseteq \mathbb{R}$ be a closed interval (this includes the cases when $a = -\infty$ or $b = \infty$, or both). The mapping $x\colon D \to \mathbb{R}^n$ is called m-times *piecewise continuously differentiable* if and only if there is a finite partition (ordered set) $\{a = t_0 < t_1 < \ldots < t_p = b\}$ of D, such that x is m-times continuously differentiable on each interval (t_i, t_{i+1}) with *càdlàg* (« continue à droite, limite à gauche ») derivative: Everywhere on D, the function x and each of its derivatives $x^{(k)}$ are right continuous and have left limits.

More precisely, for all $i = 0, \ldots, p-1$ and for all $k = 0, \ldots, m$ exist the left limits

$$\lim_{\substack{t \nearrow t_{i+1} \\ t \in (t_i, t_{i+1})}} x^{(k)}(t) \tag{3.1}$$



(a) Piecewise continuous function.

(b) Not allowed!

Figure 3.1: Examples to Definition 3.1.

as well as the right limits

$$\lim_{\substack{t \searrow t_i \\ t \in (t_i, t_{i+1})}} x^{(k)}(t) = x^{(k)}(t_i) \tag{3.2}$$

which additionally coincide with the value of $x^{(k)}$ at this knot t_i. The same holds for these limits inside the intervals, i.e. for $t \to s \in (t_i, t_{i+1})$. Hence x can have an isolated point only in the right interval limit b.

In the case $m = 0$, we say x is *piecewise continuous*. For a compact $D \subset \mathbb{R}$ (this excludes the cases with $\pm\infty$), we denote by $C^m_{\text{pw}}(D, \mathbb{R}^n)$ the set of *m-times piecewise continuously differentiable functions* on D mapping to $\mathbb{R}^n$, and respectively, by $C^0_{\text{pw}}(D, \mathbb{R}^n)$ the set of *piecewise continuous functions* on D.

The supremum norm $\|\cdot\|_{\sup}$ of the Banach space of continuous functions on the compactum D can be extended to $C^m_{\text{pw}}(D, \mathbb{R}^n)$, since each element consists of a finite number of continuous parts.

In the following, when we talk about *piecewise continuous* and *piecewise continuously differentiable*, we refer to it in the sense of Definition 3.1. Let us note some basic observations which will be used subsequently.

Lemma 3.2. *The composition of a continuous (outer) and a piecewise continuous function (inner) is again piecewise continuous with the same partition.*

Proof. The limits (3.1) and (3.2) exist, because they commute with the continuous outer function and exist for the piecewise-continuous inner function. □

Lemma 3.3. *A piecewise continuous function is (Riemann) integrable.*

Proof. This proof is usually given in every standard analysis book, see for example [Rud76, Theorem 6.10] or [Gat12, Example 11.16b]. □

The following lemma generalizes the fundamental theorem of calculus to piecewise continuous derivatives.

Lemma 3.4. *Let $F \in C^0([a,b]) \cap C^1_{\mathrm{pw}}([a,b])$ with piecewise derivative f. Then*

$$F(t) - F(a) = \int_a^t f(s)\,\mathrm{d}s$$

for all $t \in [a,b]$.

Proof. Let $\{a = t_0 < \ldots < t_p = b\}$ the partition for f. On each compact interval $[t_{i-1}, t_i]$, f is piecewise continuous and hence integrable (Lemma 3.3).

By precondition is F differentiable on $[t_{i-1}, \zeta]$ with $F' = f$ for all $\zeta \in (t_{i-1}, t_i)$. For that reason, the fundamental theorem of calculus (cf. standard analysis literature, e.g. [Gat12; Rud76]) yields

$$\int_{t_{i-1}}^{\zeta} f(s)\,\mathrm{d}s = F(\zeta) - F(t_{i-1})$$

and by the continuity of F that

$$\int_{t_{i-1}}^{t_i} f(s)\,\mathrm{d}s = \lim_{\zeta \to t_i} \int_{t_{i-1}}^{\zeta} f(s)\,\mathrm{d}s = \lim_{\zeta \to t_i} F(\zeta) - F(t_{i-1}) = F(t_i) - F(t_{i-1})$$

For any $t \in [a,b]$, there is a $k \in \{1, \ldots, p\}$ such that $t \in [t_{k-1}, t_k)$ (in the case $t = b$, set $k = p$), summation over $i = 1, \ldots, k$ yields the telescoping series

$$F(t) - F(a) = \sum_{i=1}^{k-1} \int_{t_{i-1}}^{t_i} f(s)\,\mathrm{d}s + \int_{t_{k-1}}^{t} f(s)\,\mathrm{d}s$$

what is by the additivity of the integral equivalent to

$$F(t) - F(a) = \int_a^t f(s)\,\mathrm{d}s.$$

□

3.2 Definition of DDE

There are different possibilities to define delay differential equations, depending on what application one has in mind. We restrict to a class adapted to our needs and often found in literature (see for example [Rou05]) and which covers a wide range of applications.

Definition 3.5 (Delay Differential Equation)**.** Given a function $f \colon \mathbb{R} \times \mathbb{R}^n \times \ldots \times \mathbb{R}^n \to \mathbb{R}^n$ and a set of time delays $\{\tau_j \in \mathbb{R} \mid 0 < \tau_1 < \ldots < \tau_k\}$, a functional equation of the form

$$x'(t) = f(t, x(t), x(t - \tau_1), \ldots, x(t - \tau_k)) \tag{3.3}$$

is called (first order) *delay differential equation* (DDE) with *multiple constant, discrete delays* τ_j. It is said to be *autonomous*, if its right hand side f is independent of t and *pure*, if the right hand side only depends on $x(t-\tau_j)$ but not on $x(t)$. We define its *maximal* and *minimal delay* as $\tau_{\max} \overset{\text{def}}{=} \tau_k$ and $\tau_{\min} \overset{\text{def}}{=} \tau_1$, respectively.

A DDE can be equipped with an *initial condition* $x_\sigma \colon [\sigma-\tau_{\max}, \sigma] \to \mathbb{R}^n$. It specifies the initial state, i.e. the values of x, on which the right hand side depends, starting from $t = \sigma$. Together, such a pair is called *initial value problem* (IVP):

$$\begin{cases} x'(t) = f(t, x(t), x(t-\tau_1), \ldots, x(t-\tau_k)) & \text{for } t \geq \sigma \\ x(t) = x_\sigma(t) & \text{for } t \in [\sigma - \tau_{\max}, \sigma] \end{cases} \tag{3.4}$$

Definition 3.6 (Solution of DDE)**.** A function $x \colon [\sigma - \tau_{\max}, \sigma + T] \to \mathbb{R}^n$ is called *(local) solution* of the initial value problem (3.4), if and only if there exists a $T > 0$ such that x obeys the initial condition

$$x(t) = x_\sigma(t) \quad \text{for } t \in [\sigma - \tau_{\max}, \sigma]$$

and x is continuous and piecewise continuously differentiable on $[\sigma, \sigma + T]$, fulfilling

$$x'(t) = f(t, x(t), x(t-\tau_1), \ldots, x(t-\tau_k))$$

on each (open) interval (t_i, t_{i+1}) of its partition $\{\sigma = t_0 < \ldots < t_p = \sigma + T\}$. If the function x is a solution for all $T > 0$, it is called *global.*

The piecewise continuity of the derivative means that the left limits

$$\lim_{s \nearrow t_i} x'(s) = \lim_{s \nearrow t_i} f(s, x(s), x(s-\tau_1), \ldots, x(s-\tau_k))$$

exist for $i \in \{1, \ldots, p\}$ and that

$$\lim_{s \searrow t_i} x'(s) = f(t_i, x(t_i), x(t_i-\tau_1), \ldots, x(t_i-\tau_k))$$

for the right limits in t_i, $i \in \{0, \ldots, p-1\}$. This is equivalent to the fact that it holds for the *right derivative*

$$x'_+(t) \overset{\text{def}}{=} \lim_{s \searrow t} \frac{x(s) - x(t)}{s - t} = f(t, x(t), x(t-\tau_1), \ldots, x(t-\tau_k))$$

for all $t \in [\sigma, \sigma + T]$.

3.3 Method of Steps

If we restrict the IVP (3.4) onto an interval $[\sigma, \sigma + T_1]$ with $T_1 \leq \tau_{\min}$, then the values of each $x(t-\tau_j)$ is specified by the initial condition and can thus be replaced by $x_\sigma(t-\tau_j)$. We obtain an *initial value problem* for an *ordinary differential equation.* If we can solve this ordinary IVP, i.e. if we can find a solution of the ODE on $[\sigma, \sigma + T_1]$, then we can reapply this method by plugging the computed solution into the DDE and solving the resulting ODE on the interval $[\sigma + T_1, \sigma + T_2]$, where again $T_2 \leq \tau_{\min}$. As long as one can solve the resulting ordinary differential equation (for suitable f and x_σ, the existence and uniqueness of a solution is guaranteed by Picard-Lindelöf's theorem), this step can be iterated.

This method, which allows to convert a DDE into an ODE on a certain interval, eliminating the explicit dependence on the past by inserting the initial condition, is known as *method of steps*. Examples using this method are given in [Fal06].

3.4 Existence and Uniqueness of Solutions

In this section, we show that under certain conditions, we can guarantee the existence of a solution for the DDE-IVP (3.4) and that in general, it cannot have more than one.

Definition 3.7 (Lipschitz Continuity)**.** A function $f\colon \mathbb{R} \times \mathbb{R}^n \times \ldots \times \mathbb{R}^n \to \mathbb{R}^n$ is called *(locally) Lipschitz continuous* (in its j-th argument, referring to t as zeroth argument) if and only if for all $a, b \in \mathbb{R}$ and $M > 0$ there is a $L > 0$, such that

$$\|f(t, x_1, \ldots, x_j, \ldots, x_k) - f(t, x_1, \ldots, y_j, \ldots, x_k)\| \leq L\|x_j - y_j\|$$

for all $t \in [a, b]$ and $x_i, y_i \in \mathbb{R}^n$ with $\|x_i\|, \|y_i\| \leq M$, $i \in 1, \ldots, k$.

Lemma 3.8. *Finding a solution of the initial value problem* (3.4) *is equivalent to solving the integral equation*

$$\begin{cases} x(t) = x_\sigma(\sigma) + \int_\sigma^t f(s, x(s), x(s - \tau_1), \ldots, x(s - \tau_k))\,\mathrm{d}s & \text{for } t \geq \sigma \\ x(t) = x_\sigma(t) & \text{for } t \in [\sigma - \tau_{\max}, \sigma] \end{cases}$$

where $f\colon \mathbb{R} \times \mathbb{R}^n \times \ldots \times \mathbb{R}^n \to \mathbb{R}^n$ is continuous. The integral is meant to be componentwise, if f is vector-valued.

Proof. Let x be a solution of the IVP for a $T > 0$. Thus x is (by definition) piecewise continuous on $[\sigma - \tau_{\max}, \sigma]$ and continuous and piecewise continuously differentiable on $[\sigma, \sigma + T]$ with (piecewise) derivative $t \mapsto f(t, x(t), x(t - \tau_1), \ldots, x(t - \tau_k))$. By Lemma 3.4 it follows

$$x(t) = x_\sigma(\sigma) + \int_\sigma^t f(s, x(s), x(s - \tau_1), \ldots, x(s - \tau_k))\,\mathrm{d}s$$

for $t[\sigma, \sigma + T]$, since $x_\sigma(\sigma) = x(\sigma)$. This means that x fulfills the integral equation.

Conversely, let x be a solution of the integral equation on $[\sigma - \tau_{\max}, \sigma + T]$. By the fundamental theorem of calculus, x is continuous on $[\sigma, \sigma + T]$.

From the partition $\{t_0 < \ldots < t_p\}$ of x_σ, we construct a partition of $[\sigma, \sigma + T]$ by

$$\mathcal{Z} \overset{\text{def}}{=} \{\hat{t}_0 < \ldots < \hat{t}_q\} \overset{\text{def}}{=} \{\sigma, \sigma + T\} \cup \bigcup_{j=1}^{k} \bigcup_{\substack{i=1 \\ t_i + \tau_j \geq \sigma}}^{p} \{t_i + \tau_j\} \tag{3.5}$$

Let $t \in (\hat{t}_{l-1}, \hat{t}_l)$ for any $l \in \{1, \dots, q\}$. If for any $j \in \{1, \dots, k\}$ and $i \in \{0, \dots, p\}$ was $t - \tau_j = t_i$, then $t = t_i + \tau_j = \hat{t}_r$ for a $r \in \{1, \dots, p\}$ by the construction of $\mathcal{Z}$. However this would be a contradiction to the choice of t. Hence $t - \tau_j \neq t_i$ for all $j \in \{1, \dots, k\}$ and $i \in \{0, \dots, m\}$, which implies that each $s \mapsto x(s - \tau_j) = x_\sigma(s - \tau_j)$ is continuous in the point t. Thus the composition

$$s \mapsto f(s, x(s), x(s - \tau_1), \dots, x(s - \tau_k))$$

is continuous in t. The fundamental theorem of calculus states in this case that x is differentiable in the point t and that $x'(t) = f(t, x(t), x(t - \tau_1), \dots, x(t - \tau_k))$.

For the right limits it follows by the continuity of f and the existence of the limits $\lim_{t \searrow \hat{t}_l} x(t - \tau_j) = x(\hat{t}_l - \tau_j)$, that

$$\begin{aligned} \lim_{t \searrow \hat{t}_l} x'(t) &= \lim_{t \searrow \hat{t}_l} f(t, x(t), x(t - \tau_1), \dots, x(t - \tau_k)) \\ &= f(\hat{t}_l, x(\hat{t}_l), x(\hat{t}_l - \tau_1), \dots, x(\hat{t}_l - \tau_k)) \end{aligned}$$

The left limits

$$\lim_{t \nearrow \hat{t}_l} x'(t) = \lim_{t \nearrow \hat{t}_l} f(t, x(s), x(t - \tau_1), \dots, x(t - \tau_k))$$

exist for the same reason. Summarily, x is continuous and piecewise continuously differentiable on $[\sigma, \sigma + T]$ with piecewise derivative f and it obviously obeys the initial condition, i.e. $x(t) = x_\sigma(t)$ for all $t \in [\sigma - \tau, \sigma]$. □

The most important result for the considered class of delay differential equations is the following theorem. Its proof is an adaption and extension of the existence theorem (Theorem 3.7) given in [Smi10] and the proof of uniqueness in [PW10]. It essentially reduces the DDE to a piecewise ODE by applying the method of steps.

Theorem 3.9 (Existence of a unique solution)**.** *Given a continuous function* $f\colon \mathbb{R} \times \mathbb{R}^n \times \ldots \times \mathbb{R}^n \to \mathbb{R}^n$*, satisfying the local Lipschitz condition (Def. 3.7) in its first argument, consider the IVP for a delay differential equation*

$$\begin{cases} x' = f(t, x(t), x(t - \tau_1), \dots, x(t - \tau_k)) & \textit{for } t \geq \sigma \\ x(t) = x_\sigma(t - \sigma) & \textit{for } t \in [\sigma - \tau_{\max}, \sigma] \end{cases} \tag{3.6}$$

with zero-aligned initial function.

Then, for each initial condition $x_\sigma \in C^1_{\mathrm{pw}}([-\tau_{\max}, 0], \mathbb{R}^n)$ *and start time* $\sigma \in \mathbb{R}$*, there* exists *a* unique local solution *of the IVP on a time interval* $[\sigma - \tau_{\max}, \sigma + T]$*. The duration* $T > 0$ *depends on the sup-norm and the partition of the initial condition, as well as the right hand side* f*.*

Proof. Let $\{-\tau_{\max} = t_0 < \ldots < t_p = 0\}$ be the partition of x_σ. As a piecewise continuous function, the initial condition can be bounded on $[-\tau_{\max}, 0]$ by any $M \geq \|x_\sigma\|_{\sup}$.

Since f is continuous, its sup-norm admits a maximum $K > 0$ on the compact set

$$S \overset{\text{def}}{=} [\sigma, \sigma + \tau_{\min}] \times \{x \in \mathbb{R}^n \ : \ \|x\| \leq 2M\}^{k+1}$$

Let $L > 0$ be the Lipschitz constant of f for that set with respect to its first argument. We put $T \overset{\text{def}}{=} \min\{\tau_{\min}, \frac{M}{K}\}$ to restrict in the following the domain of f as integrand to S.

We construct a series $\{x_{(m)}\}_{m\in\mathbb{N}_0}$ of piecewise continuous functions, which approximates the solution of the initial value problem. Set

$$x_{(0)}(t) = \begin{cases} x_\sigma(0) & \text{for } t \in [\sigma, \sigma + T] \\ x_\sigma(t - \sigma) & \text{for } t \in [\sigma - \tau_{\max}, \sigma] \end{cases}$$

For $m \in \mathbb{N}_{>0}$ define

$$x_{(m)}(t) = \begin{cases} x_\sigma(0) + \displaystyle\int_\sigma^t f(s, x_{(m-1)}(s), x_{(m-1)}(s - \tau_1), \dots \\ \qquad\qquad\qquad\qquad \dots, x_{(m-1)}(s - \tau_k))\,\mathrm{d}s & \text{for } t \in [\sigma, \sigma + T] \\ x_\sigma(t - \sigma) & \text{for } t \in [\sigma - \tau_{\max}, \sigma] \end{cases}$$

The integral exists, because the integrand is a composition of a continuous and piecewise continuous function, which is again piecewise continuous (Lemma 3.2) and hence by Lemma 3.3 integrable and continuous on $[\sigma, \sigma + T]$.

It holds for all $m > 0$ by the definition of this sequence that

$$\left\|x_{(m)}(t) - x_{(m-1)}(t)\right\| = 0 \quad \text{for } t \in [\sigma - \tau_{\max}, \sigma].$$

We show by induction over m that it holds

$$\left\|x_{(m)}(t) - x_{(m-1)}(t)\right\| \leq \frac{K}{L} \frac{L^m (t-\sigma)^m}{m!} \quad \text{for all } t \in [\sigma, \sigma + T].$$

Let $t \in [\sigma, \sigma + T]$. Since for all $s \in [\sigma - \tau_{\max}, \sigma + T]$ obviously $\|x_{(0)}(s)\| \leq M$, the statement for $m = 1$ follows from the boundedness of f on S and the triangle inequality for integrals:

$$\|x_{(1)}(t) - x_{(0)}(t)\| = \left\|\int_\sigma^t f(s, x_{(0)}(s), x_{(0)}(s - \tau_1), \dots, , x_{(0)}(s - \tau_k))\,\mathrm{d}s\right\| \leq K(t - \sigma)$$

In the inductive step for any $m > 0$, we assume that $\|x_{(m-1)}(s)\| \leq 2M$ for all $s \in [\sigma - \tau_{\max}, \sigma + T]$, what implies that also

$$\begin{aligned} \left\|x_{(m)}(t)\right\| &\leq \|x_\sigma(0)\| + \int_\sigma^t \left\|f(s, x_{(m-1)}(s), x_{(m-1)}(s - \tau_1), \dots, x_{(m-1)}(s - \tau_k))\right\| \mathrm{d}s \\ &\leq M + K(t - \sigma) \leq M + KT \\ &\leq 2M \end{aligned} \tag{3.7}$$

with the triangle inequality and the choice of $T \leq \frac{M}{K}$.

Using this and given the second restriction on $T \leq \tau_{\min}$, it follows by the Lipschitz property of f (for its first argument) that

$$\begin{aligned}
&\left\|x_{(m+1)}(t) - x_{(m)}(t)\right\| = \\
&\quad = \left\| \int_\sigma^t f(s, x_{(m)}(s), x_{(m)}(s-\tau_1), \ldots, x_{(m)}(s-\tau_k)) \right. \\
&\qquad\qquad \left. - f(s, x_{(m-1)}(s), x_{(m-1)}(s-\tau_1), \ldots, x_{(m-1)}(s-\tau_k))\,\mathrm{d}s \right\| \\
&\quad = \left\| \int_\sigma^t f(s, x_{(m)}(s), x_\sigma(s-\tau_1-\sigma), \ldots, x_\sigma(s-\tau_k-\sigma)) \right. \\
&\qquad\qquad \left. - f(s, x_{(m-1)}(s), x_\sigma(s-\tau_1-\sigma), \ldots, x_\sigma(s-\tau_k-\sigma))\,\mathrm{d}s \right\| \\
&\qquad\qquad\qquad \leq L \int_\sigma^t \left\|x_{(m)}(s) - x_{(m-1)}(s)\right\| \mathrm{d}s \\
&\qquad\qquad\qquad \leq \frac{L^m K}{m!} \int_\sigma^t (s-\sigma)^m \,\mathrm{d}s = \frac{L^m K}{(m+1)!}(t-\sigma)^{m+1}
\end{aligned}$$

Next note that the Cauchy criterion for convergent series ([Gat12, Theorem 6.13], [Rud76, Theorem 3.22]) applied to the exponential series states that

$$\forall \varepsilon > 0 \ \exists n_0 \in \mathbb{N}_0 \ \forall m \geq \tilde{m} \geq n_0 : \sum_{i=\tilde{m}+1}^{m} \frac{(LT)^i}{i!} < \varepsilon$$

So for any $\varepsilon > 0$ exist $n_0 \in \mathbb{N}_0$ and $m \geq n_0$, such that

$$\begin{aligned}
\left\|x_{(m)}(t) - x_{(n_0)}(t)\right\| &\leq \left\|x_{(m)}(t) - x_{(m-1)}(t)\right\| + \left\|x_{(m-1)}(t) - x_{(m-2)}(t)\right\| + \\
&\qquad + \ldots + \left\|x_{(n_0+1)}(t) - x_{(n_0)}(t)\right\| \\
&\leq \frac{K}{L}\frac{L^m (t-\sigma)^m}{m!} + \frac{K}{L}\frac{L^{m-1}(t-\sigma)^{m-1}}{(m-1)!} + \\
&\qquad + \ldots + \frac{K}{L}\frac{L^{n_0+1}(t-\sigma)^{n_0+1}}{(n_0+1)!} \\
&\leq \frac{K}{L} \sum_{i=n_0+1}^{m} \frac{(LT)^i}{i!} < \varepsilon
\end{aligned}$$

for all $t \in [\sigma, \sigma+T]$, i.e. $\{x_{(m)}\}$ is a Cauchy sequence. Since each $x_{(m)}$ is continuous on $[\sigma, \sigma+T]$, this Cauchy sequence has a limit x in the Banach space $C^0([\sigma, \sigma+T], \mathbb{R}^n)$ with respect to the sup-norm.

Again, we extend the continuous x backwards to $[\sigma - \tau_{\max}, \sigma]$ with the piecewise continuous x_σ, such that $x \in C^0_{\mathrm{pw}}([\sigma - \tau, \sigma + T], \mathbb{R}^n)$.

By the continuity of the sup-norm it follows from (3.7) that

$$\|x\|_{\sup} = \lim_{m\to\infty} \left\|x_{(m)}\right\|_{\sup} \leq 2M$$

and by the Lipschitz property of f

$$\begin{aligned}&\sup_{t\in[\sigma,\sigma+T]} \|f(t,x_{(m)}(t),x_{(m)}(t-\tau_1),\dots,x_{(m)}(t-\tau_k))\\ &\qquad\qquad\qquad - f(t,x(t),x(t-\tau_1),\dots,x(t-\tau_k))\|\\ =&\sup_{t\in[\sigma,\sigma+T]} \|f(t,x_{(m)}(t),x_\sigma(t-\tau_1-\sigma),\dots,x_\sigma(t-\tau_k-\sigma))\\ &\qquad\qquad - f(t,x(t),x_\sigma(t-\tau_1-\sigma),\dots,x_\sigma(t-\tau_k-\sigma))\|\\ \leq&\sup_{t\in[\sigma,\sigma+T]} \left\|x_{(m)}(t)-x(t)\right\|\end{aligned}$$

Given this bound, the uniform convergence (convergence in sup-norm) of $x_{(m)} \to x$, implies the uniform convergence

$$f(s,x_{(m)}(s),x_{(m)}(s-\tau_1),\dots,x_{(m)}(s-\tau_k)) \xrightarrow{m\to\infty} f(s,x(s),x(s-\tau_1),\dots,x(s-\tau_k))$$

and hence we can commute the integral and the limit process in

$$\begin{aligned}x(t) &= \lim_{m\to\infty} x_{(m+1)}\\ &= x_\sigma(0) + \lim_{m\to\infty}\int_\sigma^t f(s,x_{(m)}(s),x_{(m)}(s-\tau_1),\dots,x_{(m)}(s-\tau_k))\,\mathrm{d}s\\ &= x_\sigma(0) + \int_\sigma^t f(s,x(s),x(s-\tau_1),\dots,x(s-\tau_k))\,\mathrm{d}s\end{aligned}$$

where $t \in [\sigma,\sigma+T]$. It follows that x solves the integral equation (3.4), which by Lemma 3.8 proves the existence of a solution to the DDE fulfilling the initial condition.

It remains to show the uniqueness of a solution. Let x and $\bar{x}$ be two solutions of the IVP on $[\sigma,\sigma+T]$, coinciding on $[\sigma-\tau_{\max},\sigma]$. By Lemma 3.8 they are equivalent to solutions of the integral equations

$$x(t) = x_\sigma(0) + \int_\sigma^t f(s,x(s),x(s-\tau_1),\dots,x(s-\tau_k))\,\mathrm{d}s$$

and

$$\bar{x}(t) = x_\sigma(0) + \int_\sigma^t f(s,\bar{x}(s),\bar{x}(s-\tau_1),\dots,\bar{x}(s-\tau_k))\,\mathrm{d}s$$

We put $M \overset{\text{def}}{=} \max\{\|x\|_{\sup},\|\bar{x}\|_{\sup}\}$. Let $L > 0$ be the Lipschitz constant for f on the set $S \overset{\text{def}}{=} [\sigma, \sigma+\tau_{\min}] \times \{x \in \mathbb{R}^n \ : \ \|x\| \le M\}^{k+1}$. For $t \in [\sigma, \sigma+T]$, we set

$$\begin{aligned}
\rho(t) \overset{\text{def}}{=} \|x(t)-\bar{x}(t)\| \le \int_\sigma^t & \|f(s, x(s), x(s-\tau_1), \dots, x(s-\tau_k)) \\
& \qquad - f(s, \bar{x}(s), \bar{x}(s-\tau_1), \dots, \bar{x}(s-\tau_k))\| \,\mathrm{d}s \\
= \int_\sigma^t & \|f(s, x(s), x_\sigma(s-\tau_1-\sigma), \dots, x_\sigma(s-\tau_k-\sigma)) \\
& \qquad - f(s, \bar{x}(s), x_\sigma(s-\tau_1-\sigma), \dots, x_\sigma(s-\tau_k-\sigma))\| \,\mathrm{d}s \\
\le L \int_\sigma^t & \|x(s)-\bar{x}(s)\| \,\mathrm{d}s = L \int_\sigma^t \rho(s) \,\mathrm{d}s \\
= L \int_\sigma^t & \mathrm{e}^{-\alpha s} \rho(s) \mathrm{e}^{\alpha s} \,\mathrm{d}s \le L \sup_{s \in [\sigma, \sigma+T]} \left(\mathrm{e}^{-\alpha s}\rho(s)\right) \int_\sigma^t \mathrm{e}^{\alpha s} \,\mathrm{d}s \\
\le \frac{L}{\alpha} & \mathrm{e}^{\alpha t} \sup_{s \in [\sigma, \sigma+T]} \left(\mathrm{e}^{-\alpha s}\rho(s)\right)
\end{aligned}$$

The continuity of x and $\bar{x}$ also entails the continuity of ρ. Choosing $\alpha = 2L$ and multiplying with $\mathrm{e}^{-\alpha t} > 0$ leads to

$$\mathrm{e}^{-2Lt}\rho(t) \le \frac{1}{2} \sup_{s \in [\sigma, \sigma+T]} \left(\mathrm{e}^{-2Ls}\rho(s)\right)$$

for all $t \in [\sigma, \sigma+T]$. Hence

$$0 \le \sup_{t \in [\sigma, \sigma+T]} \left(\mathrm{e}^{-2Lt}\rho(t)\right) \le \frac{1}{2} \sup_{s \in [\sigma, \sigma+T]} \left(\mathrm{e}^{-2Ls}\rho(s)\right)$$

That is only possible if $\rho(t) = 0$ for all $t \in [\sigma, \sigma+T]$, what means $x(t) = \bar{x}(t)$. □

The solution of an DDE-IVP needs not to be limited to $[\sigma, \sigma+\tau_{\min}]$, as it is stated by Theorem 3.9. Like in the method of steps, the reasoning of the last proof can be iteratively reapplied to the solution obtained.

Corollary 3.10 (Continuability)**.** *The initial value problem for a delay differential equation* (3.4) *has a unique solution on* $[\sigma - \tau_{\max}, \sigma + S]$ *with* $S \ge T$.

In the following chapters, we will deal with delay differential equations having a polynomial right-hand side.

Corollary 3.11. *If f is a polynomial over t, $x(t)$ and $x(t-\tau_j)$, then there exists a unique solution to the initial value problem* (3.4) *with delay differential equation having f as right-hand side and a given piecewise continuous initial condition.*

Proof. As a polynomial, f is continuously differentiable and hence locally Lipschitz. The existence of a unique solution follows from Theorem 3.9 and Corollary 3.10. □

In the following chapters, we will only consider autonomous DDEs. This allows to restrict to the case of time-independent f and initial time $\sigma = 0$. The notion of solution for an autonomous DDE can be lifted to be a trajectory γ in the state space

$$\gamma\colon [0,T] \to C^1_{\mathrm{pw}}([-\tau_{\max}, 0], \mathbb{R}^n) \tag{3.8}$$

The *state* at time $t \in [0,T]$ is a function which provides a time limited history of the past evolution. This is all information needed to determine (using the DDE) the solution at time $\tilde{t} \geq t$. This notion of solution is a *dynamical systems'* point of view which turns out to be useful later.

Other results know from ordinary differential equations can be adapted to delay differential equations, such as continuous (or even differentiable) dependence of the solution on initial data, etc. (cf. [Dad02]).

Example 3.12. Delay differential equations can often incorporate a much richer behavior than ordinary differential equations. The basic ordinary IVP

$$\begin{cases} x'(t) = -x(t) \\ x(0) = x_0 \end{cases}$$

has the solution $x(t) = x_0 \mathrm{e}^{-t}$. However the similar DDE

$$\begin{cases} x'(t) = -x(t-\tau) & t \geq 0 \\ x(t) = x_0(t) & t \in [-\tau, 0] \end{cases}$$

has a much complexer dynamics, which is shown in Figure 3.2 for $x_0 \equiv 1$. The solution can be computed as a series of polynomial pieces by the method of steps.

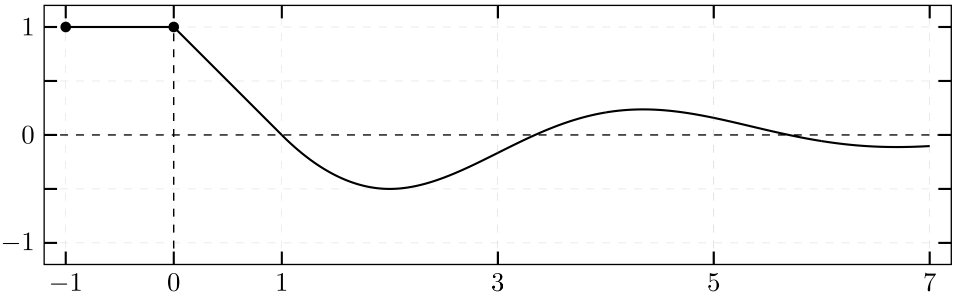


Figure 3.2: Example 3.12: Delay differential equation with constant initial condition.

4 Delay Differential Dynamic Logic

We extend classical differential dynamic logic ($\mathsf{d}\mathcal{L}$) (see e.g. [Pla12a]) with syntax, semantics, axiomatization and proof rules to support reasoning about hybrid dynamical systems with delay.

To that purpose, we allow delay differential equations in hybrid programs, which are then called *delay hybrid program* (dHP).

The definition of *delay differential dynamic logic* ($\mathsf{dd}\mathcal{L}$) provides all operators of first-order logic, as well as modal operators, in order to specify and verify reachability properties about the state of such dHPs.

In the language of $\mathsf{dd}\mathcal{L}$, we can not only model hybrid programs with DDEs in the continuous part, but also with temporal differences in the discrete fragment. As an example consider a controller which approximates a derivative by a difference quotient.

The logic $\mathsf{dd}\mathcal{L}$ is a superset of $\mathsf{d}\mathcal{L}$, i.e. in the absence of any delay, *delay differential dynamic logic* reduces to classical *differential dynamic logic.*

4.1 Syntax

Terms and formulas in $\mathsf{dd}\mathcal{L}$, as well as dHPs are defined as *words* of finite length, produced by their corresponding grammars in Backus-Naur-form (BNF).

We define by $\mathcal{V}$ be the set of *variables* and by $\mathcal{V}' \overset{\text{def}}{=} \{x' \mid x \in \mathcal{V}\}$ the corresponding set of *differential symbols.* Let $\mathcal{C} \subset \mathbb{Q}_0^-$ be the set of *constant parameters.* All three sets are supposed to be finite. We denote $\mathcal{V}[\mathcal{C}] \overset{\text{def}}{=} \{x[c] \mid x \in \mathcal{V},\ c \in \mathcal{C}\}$ as the set of *delay variables* and $\mathcal{V}'[\mathcal{C}] \overset{\text{def}}{=} \{x'[c] \mid x' \in \mathcal{V}',\ c \in \mathcal{C}\}$ as the set of *delay differentials.*

We will usually write variables as $x, y \in \mathcal{V}$ and their differential symbols as $x', y' \in \mathcal{V}'$. *Function symbols* f, g, *predicate symbols* p, q and *constant symbols* $a, b \in \mathbb{Q}$ are as in first-order logic (cf. Section 2.2).

Moreover, we write $\theta(s), \eta(s)$ for $\mathsf{dd}\mathcal{L}$ terms, $\phi(s), \psi(s)$ for $\mathsf{dd}\mathcal{L}$ formulas and α, β for dHPs. For formulas of first-order logic of real arithmetic ($\mathrm{FOL}_{\mathbb{R}}$), we use the symbols χ and φ.

Definition 4.1 (s-Terms)**.** The syntax of *terms* of *delay differential dynamic logic* is defined by the following grammar:

$$\begin{aligned}\theta(s), \eta(s) ::= {} & x[s] \mid x'[s] \mid x[c] \mid x'[c] \mid a \mid \\ & f(\theta_1(s), \dots, \theta_k(s)) \mid \theta(s) + \eta(s) \mid \theta(s) \cdot \eta(s) \mid (\theta(s))'\end{aligned}$$

where $x \in \mathcal{V}, x' \in \mathcal{V}'$ and f is a function symbol of arity k. The symbol a stands for a constant value from $\mathbb{Q}$. The constant parameters $c \in \mathcal{C}$ are not allowed to be positive.

The s-terms listed in the first line are called *atomic*, as opposed to the *composite* s-terms in the second line. S-terms generally depend on the time parameter $s \in \mathbb{R}_0^-$. This is why we write them as $\theta(s)$. If a s-term $\theta(s)$ does neither contain $x[s]$ nor $x'[s]$, we say $s \notin \theta(s)$ and abbreviate its notation to θ. Writing $\theta(b)$ means that all occurrences of s in $\theta(s)$ have been replaced with $b \in \mathbb{Q}_0^-$. Moreover, we abbreviate $x[0]$ to x and $x'[0]$ to x'. Note that $s \notin \mathcal{V} \cup \mathcal{V}' \cup \mathcal{C}$. It is a special variable symbol.

The *differential* $(\theta(s))'$ of a term $\theta(s)$ is its syntactic (total) derivative, obtained by standard differentiation rules. Lemma 4.13 shows the validity of these rules and that the result is again a s-term.

Subtraction can be defined using addition $(+)$ and multiplication $(\cdot)$, division would also be possible, if we can exclude any division by zero. The grammar allows in particular the construction of polynomial forms.

Example 4.2. Let us consider the s-term

$$\theta(s) \equiv 2x[s] + x[-\tau].$$

Setting $s = -1$ gives the term

$$\theta(-1) \equiv 2x[-1] + x[-\tau].$$

Delay differential dynamic logic uses hybrid programs with delay differential equations as system model. The grammar defining these *delayed hybrid programs* is the same as for classical HPs (cf. [Pla10b; Pla12a; Pla15a]). The difference is only given by their semantics.

Definition 4.3 (Delay Hybrid Programs)**.** The syntax of *delay hybrid programs* (dHPs) is defined by

$$\alpha, \beta ::= x := \theta \mid x' := \theta \mid ?\phi \mid \alpha \cup \beta \mid \alpha; \beta \mid \alpha^* \mid x' = \theta \,\&\, \chi$$

where α, β denote dHPs, x a variable, θ a term (possibly containing x or $x[c]$, but not $x[s]$) and ϕ a $\mathsf{dd}\mathcal{L}$ formula (not mentioning $x[s]$ or $x'[s]$). The formula χ is of $\mathrm{FOL}_{\mathbb{R}}$, containing only normal variable symbols from $\mathcal{V}$.

Note that the syntax only allows autonomous DDEs, though with multiple constant delays.

Atomic dHPs are given by instantaneous discrete *assignments*[1] $x := \theta$ and *differential assignments* $x' := \theta$, which change the value of the given variable only at the current time instant (not the past), *tests* $?\phi$, which pass only if the current state satisfies the formula ϕ and aborts the program execution if not, as well as evolutions along *delay differential equation* systems $x' = \theta \,\&\, \chi$ of an arbitrary amount of time, but restricted by the evolution domain constraint χ.

Compound dHPs combine atomic programs, and comprise *nondeterministic choices* $\alpha \cup \beta$, running either α or β, *sequential compositions* $\alpha; \beta$, executing β after α and *nondeterministic repetitions* α^*, repeating α any number of times, zero times included.

Observe that ODEs are still expressible by this syntax and that hybrid programs are hence only delayed hybrid programs with zero delay.

[1] Recall that $x := \theta$ means $x[0] := \theta$

Definition 4.4 (s-Formulas)**.** The syntax for *formulas* of *delay differential dynamic logic* is defined by the grammar

$$\begin{aligned}\phi(s), \psi(s) ::= \theta(s) = \eta(s) \mid \theta(s) \geq \eta(s) \mid p(\theta_1(s), \dots, \theta_k(s)) \mid \forall[-T)\, \phi(s) \mid \\ \neg\phi(s) \mid \phi(s) \wedge \psi(s) \mid \forall x\, \phi(s) \mid \exists x\, \phi(s) \mid [\alpha]\phi(s) \mid \langle\alpha\rangle\phi(s)\end{aligned}$$

with $\theta(s), \eta(s), \theta_1(s), \dots, \theta_k(s)$ as s-terms, p as predicate symbol, x as variable, and α as dHP.

These formulas combine connectives of propositional logic with first-order quantifiers (which all have standard meaning) and two modalities, describing *necessary* and *possible* properties.

The other comparison operators $<, \leq, >$ and logic connectives $\vee, \rightarrow, \leftrightarrow$ can be defined using $=, \geq, \wedge, \neg$ and are hence not explicitly mentioned in the grammar. Analogously is $\exists x\, \phi(s)$ expressible as $\neg\forall x\, \neg\phi(s)$ and the modal formula $[\alpha]\phi(s)$ ($\phi(s)$ holds in the state after all runs of α) by its dual $\langle\alpha\rangle\phi(s) \equiv \neg[\alpha]\neg\phi(s)$ (there is at least one state reachable by α such that $\phi(s)$ holds). The quantifiers $\forall$ and $\exists$ quantify over the elements of the state space $C^1_{\text{pw}}([-T, 0], \mathbb{R}^n)$.

Like the s-terms defined above, the s-formulas depend on a time parameter $s \in [-T, 0]$. The symbol T is a symbolic constant related to the length of the domain of the state space, which is induced by the occurrence of delay symbols. Its value is defined by the static semantics. The only way to bind the variable s in a formula $\phi(s)$ is by using $\forall[-T)\, \phi(s)$, which quantifies s over the domain of the state space, except for the current time point 0.

We write ϕ to indicate that s is not a *free variable* of $\phi(s)$ and $\phi(r)$ with $r \in [-T, 0] \subset \mathbb{R}^-_0$ to express that each term $\theta(s)$ in the formula was replaced by its corresponding $\theta(r)$, even if the term appears in the scope of a $\forall[-T)$. Sometimes we want to emphasize that a formula depends on a variable x. Then we write $\phi(x)$ and refer to x as function and element of the state space, not only the value $x[0]$ (cf. usage for $\forall$). A s-formula $\phi(s)$, which contains ($x[s]$ and) at least one $x[c]$ for any $c \in \mathcal{C}$ is called *stiff.*

Formulas of first-order logic of real arithmetic constitute a subset of $\mathsf{dd}\mathcal{L}$, i.e. every $\text{FOL}_\mathbb{R}$ formula is also a formula of delay differential dynamic logic (though not containing any delay).

Convention 4.5. The frequently appearing fact that $\phi(s)$ is not only supposed to hold for $s \in [-T, 0)$ but also in $s = 0$

$$\forall[-T)\, \phi(s) \wedge \phi(0)$$

can also be written as

$$\forall[-T]\, \phi(s)$$

For convenience, we allow the latter, abbreviated notation, which is implicitely replaced by the former, syntactically correct version.

In order to simplify notation by eliminating parentheses, we agree on the following

Convention 4.6. The operators in $\mathsf{dd}\mathcal{L}$ formulas obey the following binding priorities (from highest to lowest):

- the quantifiers $\forall, \exists$ and the modal operators $[\cdot], \langle\cdot\rangle$ bind strongest
- negation $\neg$ binds stronger than
- conjunction $\wedge$ binds stronger than
- disjunction $\vee$ binds stronger than
- implication $\rightarrow$ binds stronger than
- equivalence $\leftrightarrow$, which binds weakest.

Moreover, when a s-formula does not mention on the quantified parameter s, we can drop the quantifier $\forall[-T)$ in whose scope this formula appears.

Example 4.7. Consider the two well-formed $\mathsf{dd}\mathcal{L}$ formulas:

$$\forall[-T)\,(x + x[s] \geq 0)$$
$$\forall[-T)\,(x + x[-\tau] \geq 0)$$

The quantification over s in the second formula is vacuous and can be dropped, what leads to the equivalent formula

$$x + x[-\tau] \geq 0.$$

4.2 Dynamic Semantics

In this section, we give meaning to the syntax introduced above, by defining its semantics in a compositional way.

Following the definitions and theoretical results on the solutions of DDEs in Section 3.2, we define the *state space* in $\mathsf{dd}\mathcal{L}$ as $C^1_{\text{pw}}([-T,0],\mathbb{R}^n)$, the set of piecewise continuously differentiable functions on $[-T,0]$ (cf. Definition 3.1). This means that a variable remembers a limited part of its past evolution, what demands hence a notion of underlying time.

We denote by $\mathcal{S}$ the *set of states.* A *state* $\nu \in \mathcal{S}$ is a mapping

$$\nu\colon \mathcal{V} \cup \mathcal{V}' \to C^1_{\text{pw}}([-T,0],\mathbb{R}^n) \tag{4.1}$$

which assigns a *history* (function) to each variable and differential symbol.

By $\nu[x \mapsto y]$ we denote the state which is equal to state ν, except for the value of the variable x, which is set to $y \in C^1_{\text{pw}}([-T,0],\mathbb{R}^n)$.

Definition 4.8 (Semantics of s-terms)**.** The *semantics* of a s-term $\theta(s)$ in the state $\nu \in \mathcal{S}$ with respect to the time instant $r \in [-T,0]$ is a value in $\mathbb{R}$ and defined inductively as follows:

1. $[\![x[s]]\!]^I_{\nu,r} = \nu(x)(r)$

2. $[\![x'[s]]\!]^I_{\nu,r} = \nu(x')(r) \overset{\text{def}}{=} \lim_{t \searrow r} \frac{\nu(x)(t)-\nu(x)(r)}{t-r}$ (except in $r = 0$)

3. $[\![x[c]]\!]^I_{\nu,r} = \nu(x)(c)$

4. $[\![x'[c]]\!]^I_{\nu,r} = \nu(x')(c) \overset{\text{def}}{=} \lim_{t \searrow c} \frac{\nu(x)(t)-\nu(x)(c)}{t-c}$ (except in $c = 0$)

5. $[\![a]\!]^I_{\nu,r} = a$

6. $[\![f(\theta_1(s), \ldots, \theta_k(s))]\!]^I_{\nu,r} = I(f)([\![\theta_1(s)]\!]^I_{\nu,r}, \ldots, [\![\theta_k(s)]\!]^I_{\nu,r})$

7. $[\![\theta(s) + \eta(s)]\!]^I_{\nu,r} = [\![\theta(s)]\!]^I_{\nu,r} + [\![\eta(s)]\!]^I_{\nu,r}$

8. $[\![\theta(s) \cdot \eta(s)]\!]^I_{\nu,r} = [\![\theta(s)]\!]^I_{\nu,r} \cdot [\![\eta(s)]\!]^I_{\nu,r}$

9. $$[\![(\theta(s))']\!]^I_{\nu,r} = \sum_{x[c] \in \mathcal{V}[\mathcal{C}]} \nu(x')(c) \frac{\partial [\![\theta(s)]\!]^I_{\nu,r}}{\partial x[c]} + \nu(x')(r) \frac{\partial [\![\theta(s)]\!]^I_{\nu,r}}{\partial x[s]}$$

where $c \in \mathbb{Q}^-_0$ is a non-positive rational number and $a \in \mathbb{Q}$. Rationals have their natural meaning.

The meaning of the variable and differential symbols is determined by the state. Additionally, the value of a differential symbol has to coincide with the right derivative of the corresponding variable, except in $r = 0$, where they can mismatch.

The meaning of the differential of an arbitrary term is the total derivative of its value with respect to the underlying continuous time. As a composition of smooth functions is $[\![\theta(s)]\!]^I_{\nu,r}$ smooth itself and hence these derivatives exist. The sum is finite, since each term only mentions finitely many variables.

When a term θ does not mention s, its valuation is independent of $r \in [-T, 0]$. In this case we can write $[\![\theta]\!]^I_{\nu}$, dropping the r.

If one explicitly specifies values for the derivatives, one needs to take care that they are compatible with their corresponding variables.

Definition 4.9 (Semantics of s-formulas)**.** The semantics of a $\mathsf{dd}\mathcal{L}$ formula $\phi(s)$ is the subset of all states $[\![\phi(s)]\!]^I_r \subseteq \mathcal{S}$ in which $\phi(s)$ is true at the time instant $r \in [-T, 0]$. This set is given inductively by

1. $[\![\theta(s) = \eta(s)]\!]^I_r = \left\{\nu \in \mathcal{S} \;\middle|\; [\![\theta(s)]\!]^I_{\nu,r} = [\![\eta(s)]\!]^I_{\nu,r}\right\}$

2. $[\![\theta(s) \geq \eta(s)]\!]^I_r = \left\{\nu \in \mathcal{S} \;\middle|\; [\![\theta(s)]\!]^I_{\nu,r} \geq [\![\eta(s)]\!]^I_{\nu,r}\right\}$

3. $[\![p(\theta_1(s), \ldots, \theta_k(s))]\!]^I_r = \left\{\nu \in \mathcal{S} \;\middle|\; \left([\![\theta_1(s)]\!]^I_{\nu,r}, \ldots, [\![\theta_k(s)]\!]^I_{\nu,r}\right) \in I(p)\right\}$

4. $[\![\neg\phi(s)]\!]^I_r = \left([\![\phi(s)]\!]^I_r\right)^{\mathsf{C}} = \mathcal{S} \setminus [\![\phi(s)]\!]^I_r$

5. $[\![\phi(s) \land \psi(s)]\!]_r^I = [\![\phi(s)]\!]_r^I \cap [\![\psi(s)]\!]_r^I$

6. $[\![\forall[-T]\, \phi(s)]\!]_r^I = \left\{\nu \in \mathcal{S} \,\middle|\, \forall \tilde{r} \in [-T, 0) \,:\, \nu \in [\![\phi(s)]\!]_{\tilde{r}}^I\right\}$

7. $[\![\forall x\, \phi(s)]\!]_r^I = \left\{\nu \in \mathcal{S} \,\middle|\, \nu[x \mapsto y] \in [\![\phi(s)]\!]_r^I \text{ for all } y \in C^1_{\text{pw}}([-T, 0], \mathbb{R}^n)\right\}$

8. $[\![\exists x\, \phi(s)]\!]_r^I = \left\{\nu \in \mathcal{S} \,\middle|\, \nu[x \mapsto y] \in [\![\phi(s)]\!]_r^I \text{ for some } y \in C^1_{\text{pw}}([-T, 0], \mathbb{R}^n)\right\}$

9. $[\![[\alpha]\phi(s)]\!]_r^I = \left\{\nu \in \mathcal{S} \,\middle|\, \omega \in [\![\phi(s)]\!]_r^I \text{ for all } \omega \text{ such that} (\nu, \omega) \in \rho(\alpha)\right\},$
$= \left\{\nu \in \mathcal{S} \,\middle|\, \forall \omega \in \mathcal{S} \,:\, (\nu, \omega) \in \rho(\alpha) \Rightarrow \omega \in [\![\phi(s)]\!]_r^I\right\}$

10. $[\![\langle\alpha\rangle\phi(s)]\!]_r^I = \left\{\nu \in \mathcal{S} \,\middle|\, \omega \in [\![\phi(s)]\!]_r^I \text{ for some } \omega \text{ such that} (\nu, \omega) \in \rho(\alpha)\right\}$
$= \left\{\nu \in \mathcal{S} \,\middle|\, \exists \omega \in \mathcal{S} \,:\, (\nu, \omega) \in \rho(\alpha) \land \omega \in [\![\phi(s)]\!]_r^I\right\}$

The relation ρ is defined subsequently. The fact that formula $\phi(s)$ is true in state ν under the interpretation I at past time instant $r \in [-T, 0]$, i.e. $\nu \in [\![\phi(s)]\!]_r^I$ can also be written as $I, \nu, r \models \phi(s)$. A formula $\phi(s)$ is called valid, written as $\models \phi(s)$, if and only if $\phi(s)$ is true in all states, for all $r \in [-T, 0]$ and under all interpretations.

As in classic first-order logic, the interpretation of a predicate symbol of arity n is a relation $I(p) \subseteq \mathbb{R}^n$.

Lemma 4.10 (Time Barcan formula)**.** *The box modality and the s-quantification commute*

$$[\![\forall[-T]\, [\alpha]\phi(s)]\!]_r^I = [\![[\alpha](\forall[-T]\, \phi(s))]\!]_r^I$$

Proof. Since $\forall x \,:\, (p \Rightarrow q(x))$ iff $p \Rightarrow \forall x \,:\, q(x)$, it holds

$$\begin{aligned}
&[\![\forall[-T]\, [\alpha]\phi(s)]\!]_r^I = \\
&\qquad = \left\{\nu \in \mathcal{S} \,\middle|\, \forall \tilde{r} \in [-T, 0) \,:\, \forall \omega \in \mathcal{S} \,:\, \Big((\nu, \omega) \in \rho(\alpha) \Rightarrow \omega \in [\![\phi(s)]\!]_{\tilde{r}}^I\Big)\right\} \\
&\qquad = \left\{\nu \in \mathcal{S} \,\middle|\, \forall \omega \in \mathcal{S} \,:\, \forall \tilde{r} \in [-T, 0) \,:\, \Big((\nu, \omega) \in \rho(\alpha) \Rightarrow \omega \in [\![\phi(s)]\!]_{\tilde{r}}^I\Big)\right\} \\
&\qquad = \left\{\nu \in \mathcal{S} \,\middle|\, \forall \omega \in \mathcal{S} \,:\, \Big((\nu, \omega) \in \rho(\alpha) \Rightarrow \forall \tilde{r} \in [-T, 0) \,:\, \omega \in [\![\phi(s)]\!]_{\tilde{r}}^I\Big)\right\} \\
&\qquad = \left\{\nu \in \mathcal{S} \,\middle|\, \forall \omega \in \mathcal{S} \,:\, \Big((\nu, \omega) \in \rho(\alpha) \Rightarrow \omega \in [\![\forall[-T]\, \phi(s)]\!]_r^I\Big)\right\} \\
&\qquad = [\![[\alpha](\forall[-T]\, \phi(s))]\!]_r^I
\end{aligned}$$

□

Example 4.11. The diamond modality does not commute with the s-quantification

$$[\![\forall[-T]\, \langle\alpha\rangle\phi(s)]\!]_r^I \neq [\![\langle\alpha\rangle\forall[-T]\, \phi(s)]\!]_r^I$$

as shown by the following counter example for the state $\nu(x)(r) = 0$, $\forall r \in [-T, 0]$:

$$\nu \in [\![\forall[-T]\, \langle x' = 1\rangle (x[s] = 1)]\!]^I$$
$$\nu \notin [\![\langle x' = 1\rangle \forall[-T]\, (x[s] = 1)]\!]^I$$

Definition 4.12 (Transition semantics of dHPs)**.** The interpretation of a dHP is given by a binary *reachability relation* $\rho(\alpha) \subseteq \mathcal{S} \times \mathcal{S}$ between states:

1. $\rho(x := \theta) = \left\{ (\nu, \omega) \;\middle|\; \omega = \nu \text{ except } \omega(x) = \left(r \mapsto \begin{cases} [\![\theta(s)]\!]^I_{\nu,r} & r = 0 \\ \nu(x)(r) & r \in [-T, 0) \end{cases} \right) \right\}$
2. $\rho(x' := \theta) = \left\{ (\nu, \omega) \;\middle|\; \omega = \nu \text{ except } \omega(x') = \left(r \mapsto \begin{cases} [\![\theta(s)]\!]^I_{\nu,r} & r = 0 \\ \nu(x')(r) & r \in [-T, 0) \end{cases} \right) \right\}$
3. $\rho(?\phi) = \left\{ (\nu, \nu) \;\middle|\; \nu \in [\![\phi]\!]^I \right\}$
4. $\rho(\alpha \cup \beta) = \rho(\alpha) \cup \rho(\beta)$
5. $\rho(\alpha; \beta) = \{(\nu, \omega) \mid (\nu, \mu) \in \rho(\alpha), (\mu, \omega) \in \rho(\beta)\}$
6. $\rho(\alpha^*) = \bigcup_{m \in \mathbb{N}_0} \rho(\alpha^m)$ with $\alpha^{m+1} \equiv (\alpha^m; \alpha)$ and $\alpha^0 \equiv (?\mathit{true})$
7. $\rho(x' = \theta \,\&\, \chi) = \{(\nu, \omega) \mid \forall \zeta \in [0, R] : \gamma(\zeta) \in [\![x' = \theta \wedge \chi]\!]^I_r$ and $\nu = \gamma(0)$ on $\{x'[0]\}^\complement$ and $\omega = \gamma(r)$ for a $\gamma\colon [0, R] \to \mathcal{S}\}$, i.e. there exists a $R \geq 0$ and a trajectory $\gamma\colon [0, R] \to \mathcal{S}$, which fulfills $\gamma(\zeta)(x')(r) \overset{\text{def}}{=} \frac{\mathrm{d}\gamma(t)(x)(r)}{\mathrm{d}t}(\zeta) = [\![\theta]\!]^I_{\gamma(\zeta+r)}$ and satisfies χ for all $r \in [-\min\{\zeta, T\}, 0]$. For $r \in \big[-T, -\min\{\zeta, T\}\big)$ it holds $\gamma(\zeta)(x)(r) = \nu(x)(r + \zeta)$ for all variables x.

The semantics of a delay differential equation is motivated by the definition of a solution for a DDE-IVP (cf. Definition 3.6), following the evolution for a nondeterministic period of time, as long as the evolution domain constraint holds.

The initial value $\nu(x')(0)$ may not be compatible with the actual derivative of x. Following the evolution of a DDE, even for $R = 0$, sets it properly, such that the rate of change of the values of a variable coincide with the values of the differential symbol.

For the *discrete assignment*, we only allow the values at the current time instant to be changed. A functional assignment would essentially allow to rewrite history, which is not permitted.

The jump behavior caused by discrete assignments is the actual reason why we need to consider piecewise continuous evolutions, as defined in Def. 3.1.

Time is implicit and usually not revealed. If it is explicitly needed, a clock variable t can be introduced by $t' = 1$.

As a $\mathrm{FOL}_{\mathbb{R}}$ formula, χ do not contain any delayed variables and thus only depends on the values at the current time instant (and not on the entire interval $[-T, 0)$).

Lemma 4.13 (Derivatives)**.** *Standard analysis derivation rules also hold in the semantics of* d$\mathcal{L}$ *terms, i.e. the following equations are valid* d$\mathcal{L}$ *formulas*

$$
\begin{align}
(x[s])' &= x'[s] \tag{4.2}\\
(x[c])' &= x'[c] \tag{4.3}\\
(a)' &= 0 \tag{4.4}\\
(\theta + \eta)' &= (\theta)' + (\eta)' \tag{4.5}\\
(\theta \cdot \eta)' &= (\theta)' \cdot \eta + \theta \cdot (\eta)' \tag{4.6}\\
& \tag{4.7}
\end{align}
$$

This allows the usage of these rules on a syntactic level, what will be done in the form of axioms (see Section 5.1.1).

Proof.

$$
\begin{aligned}
[\![(x[s])']\!]^I_{\nu,r} &= \sum_{x[c]\in\mathcal{V}[\mathcal{C}]} \nu(x')(c)\frac{\partial[\![x[s]]\!]^I_{\nu,r}}{\partial x[c]} + \nu(x')(r)\frac{\partial[\![x[s]]\!]^I_{\nu,r}}{\partial x[s]}\\
&= \nu(x')(r)\frac{\partial[\![x[s]]\!]^I_{\nu,r}}{\partial x[s]} = \nu(x')(r) = [\![x'[s]]\!]^I_{\nu,r}
\end{aligned}
$$

$$
\begin{aligned}
[\![(x[c])']\!]^I_{\nu,r} &= \sum_{x[d]\in\mathcal{V}[\mathcal{C}]} \nu(x')(d)\frac{\partial[\![x[c]]\!]^I_{\nu,r}}{\partial x[d]} + \nu(x')(r)\frac{\partial[\![x[c]]\!]^I_{\nu,r}}{\partial x[s]}\\
&= \nu(x')(c)\frac{\partial[\![x[c]]\!]^I_{\nu,r}}{\partial x[c]} = \nu(x')(c) = [\![x'[c]]\!]^I_{\nu,r}
\end{aligned}
$$

$$
\begin{aligned}
[\![(a)']\!]^I_{\nu,r} &= \sum_{x[c]\in\mathcal{V}[\mathcal{C}]} \nu(x')(c)\frac{\partial[\![a]\!]^I_{\nu,r}}{\partial x[c]} + \nu(x')(r)\frac{\partial[\![a]\!]^I_{\nu,r}}{\partial x[s]}\\
&= 0
\end{aligned}
$$

$$\begin{aligned}
\llbracket(\theta(s)+\eta(s))'\rrbracket^I_{\nu,r} &= \sum_{x[c]\in\mathcal{V}[\mathcal{C}]} \nu(x')(c)\frac{\partial\llbracket\theta(s)+\eta(s)\rrbracket^I_{\nu,r}}{\partial x[c]} + \nu(x')(r)\frac{\partial\llbracket\theta(s)+\eta(s)\rrbracket^I_{\nu,r}}{\partial x[s]} \\
&= \sum_{x[c]\in\mathcal{V}[\mathcal{C}]} \nu(x')(c)\frac{\partial\left(\llbracket\theta(s)\rrbracket^I_{\nu,r}+\llbracket\eta(s)\rrbracket^I_{\nu,r}\right)}{\partial x[c]} + \nu(x')(r)\frac{\partial\left(\llbracket\theta(s)\rrbracket^I_{\nu,r}+\llbracket\eta(s)\rrbracket^I_{\nu,r}\right)}{\partial x[s]} \\
&= \sum_{x[c]\in\mathcal{V}[\mathcal{C}]} \nu(x')(c)\frac{\partial\llbracket\theta(s)\rrbracket^I_{\nu,r}}{\partial x[c]} + \nu(x')(r)\frac{\partial\llbracket\theta(s)\rrbracket^I_{\nu,r}}{\partial x[s]} \\
&+ \sum_{x[c]\in\mathcal{V}[\mathcal{C}]} \nu(x')(c)\frac{\partial\llbracket\eta(s)\rrbracket^I_{\nu,r}}{\partial x[c]} + \nu(x')(r)\frac{\partial\llbracket\eta(s)\rrbracket^I_{\nu,r}}{\partial x[s]} \\
&= \llbracket(\theta(s))'\rrbracket^I_{\nu,r} + \llbracket(\eta(s)'\rrbracket^I_{\nu,r} = \llbracket(\theta(s))' + (\eta(s))'\rrbracket^I_{\nu,r}
\end{aligned}$$

$$\begin{aligned}
\llbracket(\theta(s)\cdot\eta(s))'\rrbracket^I_{\nu,r} &= \sum_{x[c]\in\mathcal{V}[\mathcal{C}]} \nu(x')(c)\frac{\partial\llbracket\theta(s)\cdot\eta(s)\rrbracket^I_{\nu,r}}{\partial x[c]} + \nu(x')(r)\frac{\partial\llbracket\theta(s)\cdot\eta(s)\rrbracket^I_{\nu,r}}{\partial x[s]} \\
&= \sum_{x[c]\in\mathcal{V}[\mathcal{C}]} \nu(x')(c)\frac{\partial\left(\llbracket\theta(s)\rrbracket^I_{\nu,r}\cdot\llbracket\eta(s)\rrbracket^I_{\nu,r}\right)}{\partial x[c]} + \nu(x')(r)\frac{\partial\left(\llbracket\theta(s)\rrbracket^I_{\nu,r}\cdot\llbracket\eta(s)\rrbracket^I_{\nu,r}\right)}{\partial x[s]} \\
&= \sum_{x[c]\in\mathcal{V}[\mathcal{C}]} \nu(x')(c)\frac{\partial\llbracket\theta(s)\rrbracket^I_{\nu,r}}{\partial x[c]}\llbracket\eta(s)\rrbracket^I_{\nu,r} + \nu(x')(r)\frac{\partial\llbracket\theta(s)\rrbracket^I_{\nu,r}}{\partial x[s]}\llbracket\eta(s)\rrbracket^I_{\nu,r} \\
&+ \sum_{x[c]\in\mathcal{V}[\mathcal{C}]} \nu(x')(c)\frac{\partial\llbracket\eta(s)\rrbracket^I_{\nu,r}}{\partial x[c]}\llbracket\theta(s)\rrbracket^I_{\nu,r} + \nu(x')(r)\frac{\partial\llbracket\eta(s)\rrbracket^I_{\nu,r}}{\partial x[s]}\llbracket\theta(s)\rrbracket^I_{\nu,r} \\
&= \llbracket(\theta(s))'\rrbracket^I_{\nu,r}\cdot\llbracket\eta(s)\rrbracket^I_{\nu,r} + \llbracket\theta(s)\rrbracket^I_{\nu,r}\cdot\llbracket(\eta(s))'\rrbracket^I_{\nu,r} \\
&= \llbracket(\theta(s))'\cdot\eta(s) + \theta(s)\cdot(\eta(s))'\rrbracket^I_{\nu,r}
\end{aligned}$$

□

Definition 4.14. We define by

$$\mathcal{C}_\theta \overset{\text{def}}{=} \{c\in\mathcal{C} \mid \exists x\in\mathcal{V} :\ x[c]\in\theta(s)\}$$

the set of constant parameter symbols occurring in the s-term $\theta(s)$.

Note that this set does not contain s, since it is, as a special purpose symbol, not in $\mathcal{C}$.

Definition 4.15 (Sampled trajectory)**.** Since a s-term $\theta(s)$ only comprises a finite number of atomic terms, its valuation can also be seen as a mapping

$$\llbracket\theta(s)\rrbracket^I : \mathbb{R}^{|\mathcal{K}|} \to \mathbb{R}$$

from the concrete values for each element of the finite set $\mathcal{K} \overset{\text{def}}{=} \mathcal{V}[\mathcal{C}_\theta] \cup \mathcal{V}'[\mathcal{C}_\theta] \cup \{x[s], x'[s]\}$ into the reals, if we assign a fixed $r \in [-T, 0]$ to s.

This gives rise to the definition of the *sampled trajectory* $\hat{\gamma}^r_\theta \colon [0, R] \to \mathbb{R}^{|\mathcal{K}|}$ for a fixed $r \in [-T, 0]$ and s-term $\theta(s)$, which looks in the case $\mathcal{V} = \{x\}$ like

$$\hat{\gamma}^r_\theta(t) \overset{\text{def}}{=} \begin{pmatrix} \gamma(t)(x)(c_1) \\ \vdots \\ \gamma(t)(x)(c_n) \end{pmatrix}$$

The following lemma is the DDE counterpart to the differential lemma for ODEs [Pla15a] and shows the consistency of the semantics for differentials with the semantics of the evolution of a delay differential equation. This means that along a DDE, the values of differential symbols coincide with the time derivative of the value of the corresponding variable.

Lemma 4.16 (Differential Lemma)**.** *The value of a s-term $\eta(s)$ along a trajectory $\gamma \colon [0, R] \to \mathcal{S}$ satisfying a DDE for any duration $R > 0$, i.e. $I, \gamma \models (x' = \theta \wedge \chi)$, is piecewise continuously differentiable and for all $\zeta \in [0, R]$ and $r \in [-T, 0]$ it holds:*

$$[\![(\eta(s))']\!]^I_{\gamma(\zeta), r} = \frac{\mathrm{d}[\![\eta(s)]\!]^I_{\gamma(t), r}}{\mathrm{d}t}(\zeta)$$

As in Definition 3.1, the derivative at a “bend point” is to be understood as right derivative.

Proof. Without loss of generality, we restrict in this proof to a single variable x. If $\eta(s)$ depends on more variables, consider the union of their partitions in the initial condition. Let $\{-T = t_0 < \ldots < t_p = 0\}$ be the partition of the initial condition $\gamma(0)(x) \in C^1_{\text{pw}}([-T, 0], \mathbb{R}^n)$.

We choose an arbitrary but fixed valuation $r \in [-T, 0]$ for s, such that the symbol s can be treated as a constant parameter, in the same way as any $c \in \mathbb{Q}^-_0$. Depending on the s-term $\eta(s)$ and the fixed r, we define a partition $\mathcal{Z}^s_\eta = \{\hat{t}_0 < \ldots < \hat{t}_q\}$ of $[0, \infty)$ (which can be limited to $[0, R]$) by

$$\mathcal{Z}^r_\eta \overset{\text{def}}{=} \bigcup_{i=0}^{m} \bigcup_{\substack{c \in \mathcal{K} \\ t_i \geq c}} \{t_i - c\} \cup \bigcup_{i=0}^{m} \bigcup_{j=1}^{k} \bigcup_{\substack{c \in \mathcal{K} \\ t_i + \tau_j \geq c}} \{t_i + \tau_j - c\}$$

where $\mathcal{K} \overset{\text{def}}{=} \mathcal{C}_\eta \cup \{0, r\}$ is the set of the constants (the interpretations of their symbols) appearing in the term η and $\tau_j \in \mathcal{C}_\theta$ the delays in the right hand side of the DDE. The set $\mathcal{Z}^r_\eta$ is finite and non-empty, since it contains at least the value 0.

We show first that $\gamma(t)(x)(c)$ is piecewise continuously differentiable in t for each $c \in \mathcal{K}$ with partition $\mathcal{Z}^r_\eta$:

Let $c \in \mathcal{K}$ and $\zeta \in (\hat{t}_l, \hat{t}_{l+1})$. Assume that $\zeta + c = t_i$ for some i. This implies $\zeta = t_i - c = \hat{t}_k$ for some k by the definition of the partition. This is not possible by the choice of ζ lying between two consecutive $\hat{t}_l$. We apply the same argumentation to the assumption $\zeta + c = t_i + \tau_j$. These contradictions show that for $\zeta \in (\hat{t}_l, \hat{t}_{l+1})$, it holds that $\zeta + c \neq t_i$ and $\zeta + c \neq t_i + \tau_j$ for all $c \in \mathcal{K}$ and for all $i \in \{0, \ldots, m\}$ and $j \in \{1, \ldots, k\}$. We now distinguish two cases:

If $\zeta + c < 0$, it holds by the definition of the DDE semantics (Definition 4.12(7)) that $\gamma(\zeta)(x)(c) = \gamma(0)(x)(\zeta + c)$, which is continuously differentiable as initial condition, if $\zeta + c \neq t_i$. Hence it follows

$$\frac{\mathrm{d}\gamma(t)(x)(c)}{\mathrm{d}t}(\zeta) = \frac{\mathrm{d}\gamma(0)(x)(r)}{\mathrm{d}r}(\zeta + c) = \gamma(0)(x')(\zeta + c) = \gamma(\zeta)(x')(c)$$

For the right limit it holds

$$\begin{aligned}\lim_{\zeta \searrow \hat{t}_l} \frac{\mathrm{d}\gamma(t)(x)(c)}{\mathrm{d}t}(\zeta) &= \lim_{\zeta \searrow \hat{t}_l} \frac{\mathrm{d}\gamma(0)(x)(r)}{\mathrm{d}r}(\zeta + c) \\ &= \gamma(0)(x')(\hat{t}_l + c)\end{aligned}$$

since $\zeta + c \in (t_i, t_{i+1})$. And analogously for the existence of the left limit for $\zeta \nearrow \hat{t}_{l+1}$

If $\zeta + c \geq 0$, then $\gamma(\zeta)(x)(c) = \gamma(\zeta + c)(x)(0)$ is differentiable in ζ with

$$\frac{\mathrm{d}\gamma(t)(x)(c)}{\mathrm{d}t}(\zeta) = \gamma(\zeta)(x')(c)$$

by the semantics of the DDEs, if $\zeta + c \neq t_i + \tau_j$.

Let $\hat{\gamma}^r_\eta$ be the η-sampled trajectory for the considered delay differential equation and the fixed r. It follows with the above results

$$\begin{aligned}\frac{\mathrm{d}[\![\eta]\!]^I_{\hat{\gamma}^r_\eta(\zeta)}}{\mathrm{d}t} &= \Big([\![\eta]\!]^I \circ \hat{\gamma}^r_\eta(\zeta)\Big)' = \nabla [\![\eta]\!]^I(\hat{\gamma}^r_\eta(\zeta)) \cdot \frac{\mathrm{d}\hat{\gamma}^r_\eta}{\mathrm{d}t}(\zeta) \\ &= \sum_{x[c] \in \mathcal{V}[\mathcal{C}_\eta]} \frac{\mathrm{d}\gamma(t)(x)(c)}{\mathrm{d}t}(\zeta) \frac{\partial [\![\eta]\!]^I_{\hat{\gamma}^r_\eta(\zeta),r}}{\partial(x[c])} \\ &= \sum_{x[c] \in \mathcal{V}[\mathcal{C}]} \gamma(\zeta)(x')(c) \frac{\partial [\![\eta]\!]^I_{\hat{\gamma}^r_\eta(\zeta),r}}{\partial(x[c])} \\ &= [\![(\eta')]\!]^I_{\hat{\gamma}^r_\eta(\zeta)}\end{aligned}$$

where each sum only consists of finitely many summands. Moreover, it holds for the right limits

$$\lim_{\zeta \searrow \hat{t}_l} \frac{\mathrm{d}[\![\eta]\!]^I_{\hat{\gamma}^r_\eta(\zeta),r}}{\mathrm{d}t} = [\![(\eta')]\!]^I_{\hat{\gamma}^r_\eta(\hat{t}_l)}$$

and the left limits for $\zeta \nearrow \hat{t}_{l+1}$ exist. □

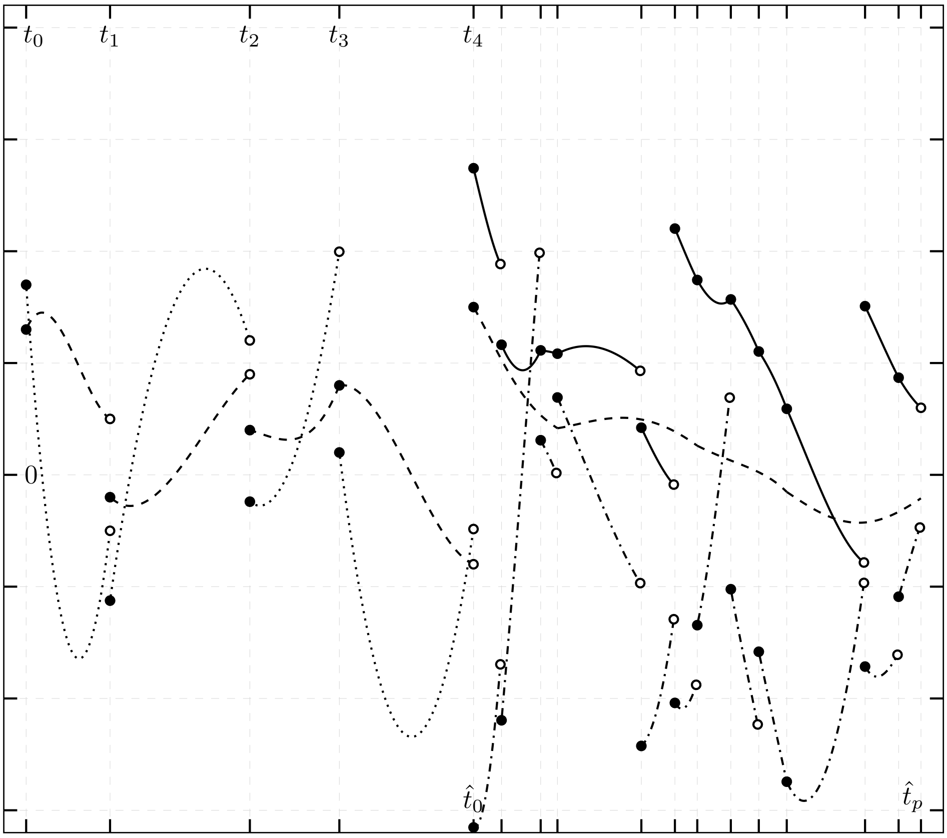


Figure 4.1: Plot for Example 4.17, initial condition and solution of DDE (dashed), derivatives (dotted), value of term (solid) and the derivative of the term (dash-dotted).

Example 4.17. As an example for the construction of the partition in Proof 4.2, consider the s-term $\eta(s) \equiv x + x[-3.5] + x[s]$ together with the DDE $x' = x[-4]$. Let

$$\mathcal{Z} = \{-4, -3.25, -2, -1.2, 0\}$$

be the partition of some initial condition. Choosing $r = -1.8$ for s, we obtain

$$\mathcal{Z}^r_\eta = \{0, 0.25, 0.6, 0.75, 1.5, 1.8, 2, 2.3, 2.55, 2.8, 3.5, 3.8, 4\}$$

when we restrict the evolution to $[0, 4]$. Figure 4.1 depicts an example for the piecewise continuous differentiability of the term's evolution, given some initial condition.

Lemma 4.18 (Differential assignment)**.** *Let $\gamma\colon [0, R] \to \mathcal{S}$ be a trajectory satisfying a DDE for any duration $R \geq 0$, i.e. $I, \gamma \models (x' = \theta \land \chi)$. Then it holds for $r \in [-T, 0]$:*

$$I, \gamma, r \models \phi(s) \iff \gamma(\zeta) \in [\![[x' := \theta]\phi(s)]\!]^I_r$$

Proof. Let $\zeta \in [0,R]$. It is $\gamma(\zeta) \in [\![x'[0] = \theta]\!]^I$ and $\gamma(\zeta) \in [\![\chi]\!]^I$, which means $\gamma(\zeta)(x')(0) = [\![\theta]\!]^I_{\gamma(\zeta)}$. By Definition 4.12(1) of the assignment's semantics, this implies $(\gamma(\zeta), \omega) \in \rho(x' := \theta)$ if and only if $\omega = \gamma(\zeta)$. Finally, this implies the equivalence

$$\begin{aligned}\gamma(\zeta) \in [\![\phi(s)]\!]^I_r &\leftrightarrow \forall \omega \in \mathcal{S} : \Big((\gamma(\zeta), \omega) \in \rho(x' := \theta) \rightarrow \omega \in [\![\phi(s)]\!]^I_r\Big) \\ &\leftrightarrow \gamma(\zeta) \in [\![[x' := \theta]\phi(s)]\!]^I_r\end{aligned}$$

□

4.3 Static Semantics

The static semantics of ddℒ formulas and dHPs defines some properties, which can be derived solely from their syntactic structure and without execution of their programs. We adapt the notion of *free* and *bound* occurrences of variables in formulas and introduce the so called *history horizon*.

4.3.1 History Horizon

Delay hybrid programs and ddℒ formulas can reference past values of variables. These values need to be specified by the state. For that reason, the lower interval bound $T \in \mathbb{R}_0^-$ of the state space domain needs to be chosen accordingly.

This bound is called *history horizon* and depends on all occurrences of $x[c]$ and $x'[c]$ in the formula and the hybrid programs it contains. The concrete value of T needs to be known in order to determine the validity of a formula, since it appears explicitly in the quantification $\forall[-T)$ for the special variable s.

Definition 4.19 (History Horizon)**.** The *history horizon* is a function which assigns to each ddℒ formula the earliest point in time it references to. It is defined inductively for s-formulas by:

$$\begin{aligned}\mathrm{HH}(\theta(s) = \eta(s)) = \mathrm{HH}(\theta(s) \geq \eta(s)) &= \max\{\mathrm{HH}(\theta(s)), \mathrm{HH}(\eta(s))\} \\ \mathrm{HH}(p(\theta_1(s), \dots, \theta_k(s))) &= \max\{\mathrm{HH}(\theta_1(s)), \dots, \mathrm{HH}(\theta_k(s))\} \\ \mathrm{HH}\big(\forall[-T)\, \phi(s)\big) &= \mathrm{HH}(\phi(s)) \\ \mathrm{HH}(\neg\phi(s)) &= \mathrm{HH}(\phi(s)) \\ \mathrm{HH}(\phi(s) \wedge \psi(s)) &= \max\{\mathrm{HH}(\phi(s)), \mathrm{HH}(\psi(s))\} \\ \mathrm{HH}(\forall x\, \phi(s)) = \mathrm{HH}(\exists x\, \phi(s)) &= \mathrm{HH}(\phi(s)) \\ \mathrm{HH}([\alpha]\phi(s)) = \mathrm{HH}(\langle\alpha\rangle\phi(s)) &= \max\{\mathrm{HH}(\alpha), \mathrm{HH}(\phi(s))\}\end{aligned}$$

depending on the *history horizon* for s-terms

$$\begin{aligned}
\mathrm{HH}(x[s]) &= 0\\
\mathrm{HH}(x'[s]) &= 0\\
\mathrm{HH}(x[c]) &= |c|\\
\mathrm{HH}(x'[c]) &= |c|\\
\mathrm{HH}(a) &= 0\\
\mathrm{HH}(f(\theta_1(s),\dots,\theta_k(s))) &= \max\{\mathrm{HH}(\theta_1(s)),\dots,\mathrm{HH}(\theta_k(s))\}\\
\mathrm{HH}(\theta(s)+\eta(s)) &= \max\{\mathrm{HH}(\theta(s)),\mathrm{HH}(\eta(s))\}\\
\mathrm{HH}(\theta(s)\cdot\eta(s)) &= \max\{\mathrm{HH}(\theta(s)),\mathrm{HH}(\eta(s))\}
\end{aligned}$$

and for dHPs

$$\begin{aligned}
\mathrm{HH}(x := \theta) &= \mathrm{HH}(\theta)\\
\mathrm{HH}(x' := \theta) &= \mathrm{HH}(\theta)\\
\mathrm{HH}(?\phi) &= \mathrm{HH}(\phi)\\
\mathrm{HH}(\alpha \cup \beta) &= \max\{\mathrm{HH}(\alpha),\mathrm{HH}(\beta)\}\\
\mathrm{HH}(\alpha;\beta) &= \max\{\mathrm{HH}(\alpha),\mathrm{HH}(\beta)\}\\
\mathrm{HH}(\alpha^*) &= \mathrm{HH}(\alpha)\\
\mathrm{HH}(x' = \theta \,\&\, \chi) &= \mathrm{HH}(\theta)
\end{aligned}$$

Example 4.20. Consider the following $\mathsf{dd}\mathcal{L}$ formula:

$$\begin{aligned}
&\forall[-T]\,(x[s]=2) \rightarrow\\
&\qquad [x := x[-3]\cdot x[-3]; x' = x[-\tau]+2x \,\&\, (x \geq 0)]\big(\forall[-T]\,0 \leq x[s] \wedge x[s] \leq x[-5]\big)
\end{aligned}$$

The history horizon needs to be set to

$$\begin{aligned}
T &= \max\{\max\{3,\tau\},\max\{0,5\}\}\\
&= \max\{\tau,5\}.
\end{aligned}$$

4.3.2 Variable Binding

Similar to $\mathsf{d}\mathcal{L}$ [Pla15a], we define *free*, *bound* and *must bound* variables. These notions are needed for the sound definition of axioms, whenever they incorporate a substitution. A variable x is bound by quantifiers of the form $\forall x$ or $\exists x$ or through discrete assignments or differential evolutions inside a modality, such as $[x := 4]$ or $\langle x' = x[-1]\rangle$. The only way to bind the special variable s is by appearing inside the scope of $\forall[-T]$.

More precisely, these notions can be defined by simultaneous induction over the syntax:

Definition 4.21 (Free variable)**.** For s-terms, we define the set $\mathrm{FV}(\theta(s)) \subseteq \mathcal{V} \cup \mathcal{V}' \cup \{s\}$ of *free variables* as the variables that occur in this term:

$$\begin{aligned}
\mathrm{FV}(x[s]) &= \{x, s\} \\
\mathrm{FV}(x'[s]) &= \{x', s\} \\
\mathrm{FV}(x[c]) &= \{x\}, \quad c \in \mathbb{Q} \\
\mathrm{FV}(x'[c]) &= \{x'\}, \quad c \in \mathbb{Q} \\
\mathrm{FV}(a) &= \emptyset \\
\mathrm{FV}\big(f(\theta_1(s), \dots, \theta_k(s))\big) &= \mathrm{FV}(\theta_1(s)) \cup \dots \cup \mathrm{FV}(\theta_k(s)) \\
\mathrm{FV}\big(\theta(s) + \eta(s)\big) = \mathrm{FV}\big(\theta(s) \cdot \eta(s)\big) &= \mathrm{FV}(\theta(s)) \cup \mathrm{FV}(\eta(s)) \\
\mathrm{FV}\big((\theta(s))'\big) &= \mathrm{FV}(\theta(s)) \cup \mathrm{FV}(\theta(s))'
\end{aligned}$$

The set $\mathrm{FV}(\phi(s)) \subseteq \mathcal{V} \cup \mathcal{V}' \cup \{s\}$ for a s-formula is defined as all its variables which appear outside the scope of quantifiers or modalities which bind it:

$$\begin{aligned}
\mathrm{FV}\big(\theta(s) = \eta(s)\big) = \mathrm{FV}\big(\theta(s) \geq \eta(s)\big) &= \mathrm{FV}(\theta(s)) \cup \mathrm{FV}(\eta(s)) \\
\mathrm{FV}\big(p(\theta_1(s), \dots, \theta_k(s))\big) &= \mathrm{FV}(\theta_1(s)) \cup \dots \cup \mathrm{FV}(\theta_k(s)) \\
\mathrm{FV}\big(\forall[-T)\, \phi(s)\big) &= \mathrm{FV}(\phi(s)) \setminus \{s\} \\
\mathrm{FV}(\neg\phi(s)) &= \mathrm{FV}(\phi(s)) \\
\mathrm{FV}(\phi(s) \wedge \psi(s)) &= \mathrm{FV}(\phi(s)) \cup \mathrm{FV}(\psi(s)) \\
\mathrm{FV}(\forall x\, \phi(s)) = \mathrm{FV}(\exists x\, \phi(s)) &= \mathrm{FV}(\phi(s)) \setminus \{x\} \\
\mathrm{FV}([\alpha]\phi(s)) = \mathrm{FV}(\langle\alpha\rangle\phi(s)) &= \mathrm{FV}(\alpha) \cup (\mathrm{FV}(\phi) \setminus \mathrm{MBV}(\alpha))
\end{aligned}$$

For a dHP α, its free variables $\mathrm{FV}(\alpha) \subseteq \mathcal{V} \cup \mathcal{V}'$ are those which are potentially read:

$$\begin{aligned}
\mathrm{FV}(x := \theta) = \mathrm{FV}(x' := \theta) &= \mathrm{FV}(\theta) \\
\mathrm{FV}(?\phi) &= \mathrm{FV}(\phi) \\
\mathrm{FV}(\alpha \cup \beta) &= \mathrm{FV}(\alpha) \cup \mathrm{FV}(\beta) \\
\mathrm{FV}(\alpha; \beta) &= \mathrm{FV}(\alpha) \cup (\mathrm{FV}(\beta) \setminus \mathrm{MBV}(\alpha)) \\
\mathrm{FV}(\alpha^*) &= \mathrm{FV}(\alpha) \\
\mathrm{FV}(x' = \theta \,\&\, \chi) &= \{x\} \cup \mathrm{FV}(\theta) \cup \mathrm{FV}(\chi)
\end{aligned}$$

Only bound variables can change their value during the execution of a delay hybrid program. For formulas, bound variables are the variables which don't need an assignment in order to determine the truth value of the formula.

Definition 4.22 (Bound variable)**.** The set $\mathrm{BV}(\phi(s)) \subseteq \mathcal{V} \cup \mathcal{V}' \cup \{s\}$ of a s-formula is defined as:

$$\begin{aligned}
\mathrm{BV}\big(\theta(s) = \eta(s)\big) = \mathrm{BV}\big(\theta(s) \geq \eta(s)\big) &= \emptyset \\
\mathrm{BV}\big(p(\theta_1(s), \dots, \theta_k(s))\big) &= \emptyset \\
\mathrm{BV}\big(\forall[-T)\, \phi(s)\big) &= \{s\} \cup \mathrm{BV}(\phi(s)) \\
\mathrm{BV}(\neg\phi(s)) &= \mathrm{BV}(\phi(s)) \\
\mathrm{BV}(\phi(s) \wedge \psi(s)) &= \mathrm{BV}(\phi(s)) \cup \mathrm{BV}(\psi(s)) \\
\mathrm{BV}(\forall x\, \phi(s)) = \mathrm{BV}(\exists x\, \phi(s)) &= \{x\} \cup \mathrm{BV}(\phi(s)) \\
\mathrm{BV}([\alpha]\phi(s)) = \mathrm{BV}(\langle\alpha\rangle\phi(s)) &= \mathrm{BV}(\alpha) \cup \mathrm{BV}(\phi(s))
\end{aligned}$$

The set of *bound variables* $\mathrm{BV}(\alpha) \subseteq \mathcal{V} \cup \mathcal{V}'$ of a dHP α are the variables which are potentially written:

$$\begin{aligned}
\mathrm{BV}(x := \theta) &= \{x\} \\
\mathrm{BV}(x' := \theta) &= \{x'\} \\
\mathrm{BV}(?\phi) &= \emptyset \\
\mathrm{BV}(\alpha \cup \beta) = \mathrm{BV}(\alpha; \beta) &= \mathrm{BV}(\alpha) \cup \mathrm{BV}(\beta) \\
\mathrm{BV}(\alpha^*) &= \mathrm{BV}(\alpha) \\
\mathrm{BV}(x' = \theta \,\&\, \chi) &= \{x, x'\}
\end{aligned}$$

Definition 4.23 (Must-bound variable)**.** The set of *must-bound variables* $\mathrm{MBV}(\alpha) \subseteq \mathcal{V} \cup \mathcal{V}'$ of a dHP α, are the variables which are written at all execution paths:

$$\begin{aligned}
\mathrm{MBV}(x := \theta) &= \mathrm{BV}(x := \theta) = \{x\} \\
\mathrm{MBV}(x' := \theta) &= \mathrm{BV}(x' := \theta) = \{x'\} \\
\mathrm{MBV}(?\phi) &= \mathrm{BV}(?\phi) = \emptyset \\
\mathrm{MBV}(\alpha \cup \beta) &= \mathrm{MBV}(\alpha) \cap \mathrm{MBV}(\beta) \\
\mathrm{MBV}(\alpha; \beta) &= \mathrm{MBV}(\alpha) \cup \mathrm{MBV}(\beta) \\
\mathrm{MBV}(\alpha^*) &= \emptyset \\
\mathrm{MBV}(x' = \theta \,\&\, \chi) &= \mathrm{BV}(x' = \theta \,\&\, \chi) = \{x, x'\}
\end{aligned}$$

Obviously, it holds $\mathrm{MBV}(\alpha) \subseteq \mathrm{BV}(\alpha)$.

5 Axiomatization and Proof Calculus

Formulas of delay differential dynamic logic allow the specification of properties of hybrid programs with delay. Their truth value, whether a such formula is true or false, is determined by the semantics. However, finding the truth value only using the definition of the semantics is unpractical and tedious. As a more powerful means for this verification task, $\mathsf{dd}\mathcal{L}$ includes a proof calculus with rules based on axioms. These allow manipulations on a syntactic level, without falling back on the semantics. Moreover, they can be implemented in software for computer-aided verification.

5.1 Axiomatization

The axiomatization for $\mathsf{dd}\mathcal{L}$ presented here is based on the $\mathsf{d}\mathcal{L}$ axiomatization, as given in [Pla12b]. It is a first-order Hilbert calculus, using *modus ponens* and $\forall$*-generalization* as a basis.

Similar to differential forms for $\mathsf{d}\mathcal{L}$ [Pla15a], we also consider a differential form axiomatization of differential equations.

The goal of transforming a $\mathsf{dd}\mathcal{L}$ formula into another formula by applying the axioms is to eventually derive a first-order formula of real arithmetic, which is decidable by *quantifier elimination.* The usage of real arithmetic is noted as ($\mathbb{R}$).

As defined by the semantics $\forall x\, \phi(s)$ and $\exists x\, \phi(s)$ quantify over the state space $C^1_{\text{pw}}([-T,0],\mathbb{R}^n)$, not only the reals. However, since we have only finitely many $x[c] \in \phi(s)$, we can introduce a quantification over $\mathbb{R}$ for each $x[c]$. The $x[s]$ are bound by $\forall[-T)$, which already quantifies over a subset of $\mathbb{R}$. This means, that we can lift the quantifier elimination of $\text{FOL}_{\mathbb{R}}$ to our needs.

If the $\mathsf{dd}\mathcal{L}$ formula ϕ can be derived by $\mathsf{dd}\mathcal{L}$ proof rules from $\mathsf{dd}\mathcal{L}$ axioms (this includes first-order logic axioms), we say that this formula is *provable* and write $\vdash \phi$.

The axioms listed in Figure 5.1 are expressed in $[\cdot]$. Axioms with the dual operator $\langle\cdot\rangle$ can be obtained using the duality relation (axiom $\langle\cdot\rangle$).

The *discrete assignment axiom* $[:=]$ substitutes $x[0]$ (alias x) by its new value θ in the formula, requirering the admissibility condition (neither x nor any variable in θ get bound by a quantifier or modality, cf. Section 4.3.2) to hold. The *test axiom* $[?]$ assumes that the test passes, because states which do not satisfy the condition are not considered anyway. The *axiom of nondeterministic choice* $[\cup]$ splits the transition into its two possibilities, demanding the validity of either. The *composition axiom* $[;]$ splits the modality into a nested modality. The *iteration axiom* $[*]$ partially unwinds a loop, which can also be used for bounded model checking. The *induction axiom* I

$$
\begin{array}{rll}
\langle\cdot\rangle & \langle\alpha\rangle\phi(s) \leftrightarrow \neg[\alpha]\neg\phi(s) & \\
[:=] & [x := \theta]\phi(s, x[0]) \leftrightarrow \phi(s, \theta) & \\
[?] & [?\psi]\phi(s) \leftrightarrow \big(\psi \rightarrow \phi(s)\big) & \\
[\cup] & [\alpha \cup \beta]\phi(s) \leftrightarrow [\alpha]\phi(s) \wedge [\beta]\phi(s) & \\
[;] & [\alpha; \beta]\phi(s) \leftrightarrow [\alpha][\beta]\phi(s) & \\
[*] & [\alpha^*]\phi(s) \leftrightarrow \phi(s) \wedge [\alpha][\alpha^*]\phi(s) & \\
\mathrm{K} & [\alpha]\big(\phi(s) \rightarrow \psi(s)\big) \rightarrow \big([\alpha]\phi(s) \rightarrow [\alpha]\psi(s)\big) & \\
\mathrm{I} & [\alpha^*]\big(\phi(s) \rightarrow [\alpha]\phi(s)\big) \rightarrow \big(\phi(s) \rightarrow [\alpha^*]\phi(s)\big) & \\
\mathrm{B} & \forall x\, [\alpha]\phi(s) \rightarrow [\alpha]\forall x\, \phi(s) & (x \notin \alpha) \\
\mathrm{V} & \phi(s) \rightarrow [\alpha]\phi(s) & (\mathrm{FV}(\phi) \cap \mathrm{BV}(\alpha) = \emptyset)
\end{array}
$$

$$
\begin{array}{rl}
\mathrm{G} & \dfrac{\phi(s)}{[\alpha]\phi(s)} \\
\mathrm{MP} & \dfrac{\phi(s) \rightarrow \psi(s) \quad \phi(s)}{\psi(s)} \\
\forall & \dfrac{\phi(s)}{\forall x\, \phi(s)}
\end{array}
$$

Figure 5.1: Delay differential dynamic logic axioms and proof rules.

can be applied when reasoning about loops with unbounded repetitions. The *modal modus ponens* axiom K and the *Barcan formula* B are taken from first-order modal logic. Axiom V is for modalities which are true for any state transition, since their condition does not depend on the variables that change.

The basic *proof rules* for the presented Hilbert calculus are *Gödel's necessitation rule* G of modal logic, as well as *modus ponens* MP and ∀-*generalization* ∀ of first-order logic.

5.1.1 Differential Axioms

Some additional axioms are added to the calculus, allowing reasoning about delay differential equations. Most are similar to corresponding d$\mathcal{L}$ axioms and listed in Figure 5.2.

The *delay differential weakening axiom* DDW internalizes that all values referenced in the state after an evolution along a DDE were either specified in the initial condition or result from the differential equation. In the latter case, they need to satisfy the evolution domain constraint. It should be pointed out, that the right hand side only demands the weaker $\forall[-T)$, as opposed to the $\forall[-T]$ appearing on the left.

$x'[c]$ and $x[c]$ are not allowed in the expression of an differential invariant, because they would lead to discontinuities. Thus differential invariants φ must be $\mathrm{FOL}_{\mathbb{R}}$ formulas.

$$
\begin{array}{rl}
c' & (a)' \;=\; 0 \\[6pt]
x[\cdot]' & (x[c])' \;=\; x'[c] \\[6pt]
 & (x[s])' \;=\; x'[s] \\[6pt]
+' & (\theta(s) + \eta(s))' \;=\; (\theta(s))' + (\eta(s))' \\[6pt]
\cdot' & (\theta(s) \cdot \eta(s))' \;=\; (\theta(s))' \cdot \eta(s) + \theta(s) \cdot (\eta(s))' \\[6pt]
\mathrm{DW} & [x' = \theta \,\&\, \chi]\chi \\[6pt]
\mathrm{DC} & \big([x' = \theta \,\&\, \chi]\phi(s) \leftrightarrow [x' = \theta \,\&\, \chi \wedge \varphi]\phi(s)\big) \leftarrow [x' = \theta \,\&\, \chi]\varphi \\[6pt]
\mathrm{DE} & [x' = \theta \,\&\, \chi]\phi(s, x, x') \leftrightarrow [x' = \theta \,\&\, \chi][x' := \theta]\phi(s, x, x') \\[6pt]
\mathrm{DI} & [x' = \theta \,\&\, \chi]\varphi \leftarrow \big(\chi \rightarrow \varphi \wedge [x' = \theta \,\&\, \chi](\varphi)'\big) \\[6pt]
\mathrm{DDW} & \big(\psi \rightarrow [x' = \theta \,\&\, \chi]\forall[-T]\,\phi(s)\big) \leftarrow \big((\psi \rightarrow \forall[-T)\,\phi(s)) \wedge \forall x\,(\chi \rightarrow \phi(0))\big) \quad (x[c] \notin \phi(s))
\end{array}
$$

Figure 5.2: Delay differential equation axioms and differential axioms.

5.1.2 Axiom of Steps

The *method of steps* presented in Section 3.3 translates into an axiom [⇥].

By introducing a fresh variable t as clock, we restrict the evolution of a delay differential equation to a duration not longer than its smallest delay $\tau_{\min}$. This evolution is then wrapped in a loop. In this case, the right hand side of the differential equation only depends on the initial state of the loop, not on its own solution. Hence the differential equation is not longer a DDE, but of *ordinary* type. Its right hand side is in general piecewise continuous. Theorem 3.9 shows the existence of a unique local solution in this case.

$$
[⇥] \quad [x' = \theta \,\&\, \chi]\phi(s) \leftrightarrow [?\chi; x' := \theta; (t := 0; t' = 1, x' = \theta \,\&\, \chi \wedge 0 \le t \le \tau_{\min})^{*}]\phi(s)
$$

where $\tau_{\min}$ is the (by magnitude) smallest delay appearing in θ.

5.2 Soundness

The following theorem is obviously fundamental for the presented theory.

Theorem 5.1 (Soundness of dd$\mathcal{L}$**).** *The dd$\mathcal{L}$ calculus is sound: every formula which is provable from dd$\mathcal{L}$ axioms by dd$\mathcal{L}$ proof rules is* valid *(true in all states), i.e.* $\vdash \phi$ *implies* $\models \phi$.

Proof. The soundness proof of most of the axioms adapted from d$\mathcal{L}$ are independent of the definition of the state space, they only reason about states without considering their structure. This is the case for [;], [∪], [∗], K, I, B, V and G, whose proof can be found in the literature to d$\mathcal{L}$ [Pla12b]. The proof of [?] additionally requires the semantics of $\neg$ being defined as complement.

[:=] It is $\nu \in [\![[x := \theta]\phi(s, x[0])]\!]^I_r$ iff $\omega \in [\![\phi(s, x[0])]\!]^I_r$ for all $(\nu, \omega) \in \rho(x := \theta)$. There exists only a unique such state $\omega \in \mathcal{S}$. For this state it holds $\omega = \nu$ except for the variable x, for which

$$\omega(x)(r) = \begin{cases} [\![\theta]\!]^I_\nu & \text{if } r = 0 \\ \nu(x)(r) & \text{if } r \in [-T, 0) \end{cases}$$

i.e. the two states coincide in the values for $x[s]$, except in $x[0]$. Hence $\nu \in [\![\phi(s, \theta)]\!]^I_r$ iff $\omega \in [\![\phi(s, x[0])]\!]^I_r$. The same holds if $\phi(x[0])$ does not depend on s.

[⇥] Let $\nu \in [\![[x' = \theta \,\&\, \chi]\phi(s)]\!]^I_r$ and $\gamma\colon [0, R] \to \mathcal{S}$ be a trajectory of duration $R \geq 0$ solving the DDE and having ν as initial condition, i.e. $\gamma(0) = \nu$ on $\{x'[0]\}^{\complement}$ and $\gamma(\zeta) \in [\![x'[0] = \theta \wedge \chi]\!]^I$ for all $\zeta \in [0, R]$. By the choice of ν it holds $\gamma(\zeta) \in [\![\phi(s)]\!]^I_r$ for all $\zeta \in [0, R]$ and $r \in [-T, 0]$.

We need to show that

$$\nu \in [\![[x' := \theta; (t := 0; t' = 1, x' = \theta \,\&\, \chi \wedge 0 \leq t \leq \tau_{\min})^*]\phi(s)]\!]^I_r.$$

If we enter the loop in the right hand side zero times, this holds since $\gamma(0) \in [\![\phi(s)]\!]^I_r$ and $\gamma(0) = \nu[x'[0] \mapsto \theta]$. If we repeated the loop m times, it holds after the last iteration that $\zeta = (m-1)\tau_{\min} + t \leq R$ and that the evolution is still restricted by χ. We know in this case that $\gamma(\zeta) \in [\![\phi(s)]\!]^I_r$ what implies the assertion. The converse implication is shown analogously.

$+'$,$\cdot'$, c',$x[\cdot]'$ These axioms are special instances of the equations proved in Lemma 4.13.

DW Proof as for d$\mathcal{L}$.

DC For a formula φ of FOL$_\mathbb{R}$, let $\nu \in [\![[x' = \theta \,\&\, \chi]\varphi]\!]^I_r$ and $\gamma\colon [0, R] \to \mathcal{S}$ be an arbitrary trajectory of duration $R \geq 0$, solving the DDE and having ν as initial condition, i.e. $\gamma(0) = \nu$ on $\{x'\}^{\complement}$ and $\gamma(\xi) \in [\![x'[0] = \theta \wedge \chi]\!]^I_r$ for all $\xi \in [0, R]$.

Suppose $\nu \in [\![[x' = \theta \,\&\, \chi]\phi(s)]\!]^I_r$, i.e. there exists a $0 \leq \bar{r} \leq R$, such that $\gamma(\zeta) \in [\![x' = \theta \wedge \chi]\!]^I_r$ for all $\zeta \in [0, \bar{r}]$ and $\gamma(\bar{r}) \in [\![\phi(s)]\!]^I_r$. Since $\zeta \leq R$ it is also $\gamma(\zeta) \in [\![\varphi]\!]^I_r$. This is equivalent to $\gamma(\zeta) \in [\![x' = \theta \wedge \chi \wedge \varphi]\!]^I_r$ and $\gamma(\zeta) \in [\![\varphi]\!]^I_r$ for all $\zeta \in [0, \bar{r}]$, which is the same as $\nu \in [\![[x' = \theta \,\&\, \chi \wedge \varphi]\phi(s)]\!]^I_r$.

DI This proof is an adaption of the d$\mathcal{L}$ proof for DI given in [Pla15a]. Without loss of generality we restrict to the case of invariants of the form $\varphi \equiv (g(x) \geq 0)$, where g is a term of FOL$_\mathbb{R}$. Then $(\varphi)' \equiv ((g(x))' \geq 0)$ (by ??).

Consider a state $\nu \in \mathcal{S}$ with $\nu \in [\![\chi \to \varphi \wedge [x' = \theta \,\&\, \chi](\varphi)']\!]^I_r$. We need to distinguish two cases. If $\nu \notin [\![\chi]\!]^I_r$, then there is no solution of the DDE and hence $\nu \in [\![[x' = \theta \,\&\, \chi]\varphi]\!]^I_r$ vacuously.

If $\nu \in [\![\chi]\!]^I_r$, then $\nu \in [\![[x' = \theta \,\&\, \chi](\varphi)']\!]^I_r$. Let $\gamma\colon [0, R] \to \mathcal{S}$ be a trajectory solving the DDE for some time $r \geq 0$, i.e. $I, \gamma \models (x' = \theta \,\&\, \chi)$. If $R = 0$ then $\nu \in [\![\chi]\!]^I_r$ since the only varaible changing its value is x', which is not contained in χ ($\mathrm{FV}(\chi) \cap \{x'\} = \emptyset$). Hence it follows from the precondition that $\nu \in [\![\varphi]\!]^I_r$ and for this reason $\nu \in [\![[x' = \theta \,\&\, \chi]\varphi]\!]^I_r$.

If $R > 0$, $\nu \in [\![[x' = \theta \,\&\, \chi](\varphi)']\!]^I_r$ implies $I, \gamma \models (\varphi)'$. By the Differential Lemma 4.16 it holds for all $\zeta \in [0, R]$

$$0 \leq [\![(g(x))']\!]^I_{\gamma(\zeta),r} = \frac{\mathrm{d}[\![g(x)]\!]^I_{\gamma(t),r}}{\mathrm{d}t}(\zeta)$$

$\mathrm{FV}(\varphi) \cap \{x'\} = \emptyset$ (means no x' in invariant) implies $\gamma(0) = \nu \in [\![g(x) \geq 0]\!]^I_r$. no $x[c]$ in invariant, hence continuous. Lemma 3.3 yields for any $z \in [0, R]$

$$[\![g(x)]\!]^I_{\gamma(z),r} = [\![g(x)]\!]^I_{\gamma(0),r} + \int_0^z \frac{\mathrm{d}[\![g(x)]\!]^I_{\gamma(t),r}}{\mathrm{d}t}(\zeta)\,\mathrm{d}\zeta \geq 0$$

hence $\gamma(z) \in [\![\varphi]\!]^I_r$ and hence $\nu \in [\![[x' = \theta \,\&\, \chi]\varphi]\!]^I_r$

DE Let $\nu \in [\![[x' = \theta \,\&\, \chi]\phi(s)]\!]^I_r$ and $\gamma\colon [0, R] \to \mathcal{S}$ be a trajectory of duration $R \geq 0$ solving the DDE and having ν as initial condition, i.e. $\gamma(0) = \nu$ on $\{x'[0]\}^{\complement}$ and $\gamma(\zeta) \in [\![x'[0] = \theta \wedge \chi]\!]^I_r$ for all $\zeta \in [0, R]$. Since $\nu \in [\![[x' = \theta \,\&\, \chi]\phi(s)]\!]^I_r$, we have $\gamma(R) \in [\![\phi]\!]^I_r$ which, by the Differential Assignment Lemma 4.18, is equivalent to $\gamma(R) \in [\![[x' := \theta]\phi(s)]\!]^I_r$. Hence $\nu \in [x' = \theta \,\&\, \chi][x' := \theta]\phi(s)$. The inverse implication is shown in the same way.

DDW Let $\nu \in [\![\psi \to \forall[-T]\,\phi(s)]\!]^I$ and $\nu \in [\![\forall x\,(\chi \to \phi(0))]\!]^I$. We distinguish two cases: if $\nu \notin [\![\psi]\!]^I$, then the implication holds vacuously. If $\nu \in [\![\psi]\!]^I$ then ν also satisfies $\forall[-T]\,\phi(s)$. We follow the given DDE along the trajectory $\gamma\colon [0, R] \to \mathcal{S}$ starting from ν and of duration $R \geq 0$, i.e. $(\nu, \gamma(\zeta)) \in \rho(x' = \theta \,\&\, \chi)$ and hence $\gamma(\zeta) \in [\![\chi]\!]^I$ for all $\zeta \in [0, R]$. Let $\omega = \gamma(R)$ be the state after the evolution. We show that $\omega \in [\![\phi(s)]\!]^I_r$ for all $r \in [-T, 0]$.

Since the formula $\phi(s)$ is demanded not to be stiff, i.e. $x[c] \notin \phi(s)$, it is $\omega \in [\![\phi(s)]\!]^I_r$ iff $\gamma(R - r) \in [\![\phi(s)]\!]^I_0$.

For $R - r < 0$ this follows since the initial state satisfies $\forall[-T]\,\phi(s)$. For $R - r \geq 0$, the satisfaction of the evolution domain constraint $\gamma(\zeta) \in [\![\chi]\!]^I_r$ implies $\gamma(\zeta) \in [\![\phi(s)]\!]^I_0$.

□

5.3 Delay Differential Induction

In order to facilitate a frequently appearing sequence of proof steps, we combine the differential cut, the differential invariant and delay differential weakening axioms to a new axiom, called *delay differential induction axiom.* The idea behind this derived axiom is that if the initial condition fulfills a certain safety condition $\phi(s)$ and (with the evolution domain constraint χ) an invariant ($\mathrm{FOL}_{\mathbb{R}}$) formula φ and if following the evolution of the DDE leads into the direction of safe values, it is sufficient that the invariant (with χ) implies the safety condition for the current time instant $\phi(0)$ in order to establish the validity of the safety condition for the entire final state after the DDE-evolution.

$$\text{DDI}\quad \frac{\psi \to \forall[-T)\,\phi(s) \quad \forall x\,\big(\chi \land \varphi \to \phi(0)\big) \quad \psi \land \chi \to \varphi \quad \psi \to [x' = \theta \,\&\, \chi](\varphi)'}{\psi \to [x' = \theta \,\&\, \chi]\forall[-T]\,\phi(s)}$$

For this rule to be sound, we demand the safety condition $\phi(s)$ not to be a stiff s-formula, i.e. $x[c] \notin \phi(s)$ for any $c \in \mathcal{C}$.

Proof. In the given Hilbert calculus, it holds:

$$\begin{array}{rc}
 & \big(\psi \to \forall[-T)\,\phi(s)\big) \land \forall x\,\big(\chi \land \varphi \to \phi(0)\big) \land (\psi \land \chi \to \varphi) \land \big(\psi \to [x' = \theta \,\&\, \chi](\varphi)'\big) \\ \hline
\text{DW} & \psi \to \big([x' = \theta \,\&\, \chi \land \varphi]\forall[-T]\,\phi(s) \land (\chi \to \varphi) \land [x' = \theta \,\&\, \chi](\varphi)'\big) \\ \hline
\text{DI} & \psi \to \big([x' = \theta \,\&\, \chi \land \varphi]\forall[-T]\,\phi(s) \land [x' = \theta \,\&\, \chi]\varphi\big) \\ \hline
\text{DC} & \psi \to [x' = \theta \,\&\, \chi]\forall[-T]\,\phi(s)
\end{array}$$

□

Example 5.2. Consider the first-order delay differential equation

$$\begin{cases} x'(t) = x(t-\tau) & t \geq \sigma \\ x(t) \geq 0 & t \in [\sigma - \tau, \sigma] \end{cases}$$

for any $\tau > 0$. Using the invariant $\varphi \equiv (x^3 \geq 0)$, we prove that the solution stays non-negative for all time $t \geq \sigma - \tau$. Since the DDE is autonomous, we can assume $\sigma = 0$.

$$\begin{array}{rc}
 & * \\ \hline
\mathbb{R} & (\forall[-T]\,x[s] \geq 0 \to x^3 \geq 0 \land 3x^2x[-\tau] \geq 0 \land \forall[-T)\,x[s] \geq 0) \land \forall x\,(x^3 \geq 0 \to x \geq 0) \\ \hline
[:=] & (\forall[-T]\,x[s] \geq 0 \to x^3 \geq 0 \land [x' := x[-\tau]](3x^2x' \geq 0) \land \forall[-T)\,x[s] \geq 0) \land \forall x\,(x^3 \geq 0 \to x \geq 0) \\ \hline
\text{DE,G} & (\forall[-T]\,x[s] \geq 0 \to x^3 \geq 0 \land [x' = x[-\tau]](3x^2x' \geq 0) \land \forall[-T)\,x[s] \geq 0) \land \forall x\,(x^3 \geq 0 \to x \geq 0) \\ \hline
\text{DDW} & \forall[-T]\,x[s] \geq 0 \to x^3 \geq 0 \land [x' = x[-\tau] \,\&\, x^3 \geq 0](\forall[-T]\,x[s] \geq 0) \land [x' = x[-\tau]](3x^2x' \geq 0) \\ \hline
\text{DI} & \forall[-T]\,x[s] \geq 0 \to [x' = x[-\tau] \,\&\, x^3 \geq 0](\forall[-T]\,x[s] \geq 0) \land [x' = x[-\tau]](x^3 \geq 0) \\ \hline
\text{DC} & \forall[-T]\,x[s] \geq 0 \to [x' = x[-\tau]](\forall[-T]\,x[s] \geq 0)
\end{array}$$

The first three axioms correspond to the derived axiom DDI. In the same way we can prove that the solution stays non-positive for all t, if the initial condition is non-positive.

6 Examples

To motivate the practical relevance of hybrid programs incorporating delayed differential equations, we present some examples of controllers modeled in $\mathsf{dd}\mathcal{L}$.

6.1 Car Following Model

The following example is adapted from [Ern09]. We want to model an acceleration-controller for a car, trying to keep a constant distance to another car driving in front.

We consider a continuous controller, though with an internal delay τ between the sensed velocities and the controller output, caused by the internal processing time of the controller.

Given the speed pattern of the leading car (n), the systems models the position and velocity of the following car $(n+1)$ depending on its controlled acceleration by:

$$\begin{cases} x''_{n+1}(t) = \alpha(x'_n(t-\tau) - x'_{n+1}(t-\tau)) \\ x'_n(t) = v(t) \end{cases}$$

The coefficient α can be seen as a comfort or sport factor, describing the strength of acceleration and deceleration applied to the car. By introducing v_{n+1} for x'_{n+1}, we reduce the system to first order.

The preconditions need to be specified for all $t \in [-\tau, 0]$. We consider feasible intervals for the initial distance between the cars, the velocity of the following car and specify a constant speed v_0 for the leading car.

$$\begin{aligned} d(t) = x_n(t) - x_{n+1}(t) &\in [D-m, D+m] \\ x'_{n+1}(t) &\in [V-l, V+l] \\ x'_n(t) &= v(t) = v_0 > 0 \end{aligned}$$

Both $\alpha > 0$ and $\tau > 0$ are considered to be constant.

Model 6.1 gives a possible implementation of the described controller as delay hybrid program, using the language of $\mathsf{dd}\mathcal{L}$ to describe its properties.

Of practical interest would be to prove the safety condition that the cars always keep a certain minimal distance, i.e. $d \geq \delta$, after having chosen a feasible α in the ctrl-part.

Simulations show that too strong reactions of the following car lead to damped oscillations. Thus, the intelligent choice of α is of interest.

Model 6.1 Example model in $\mathsf{dd}\mathcal{L}$ for the following car controller.

$$\begin{aligned}
&\text{init} \to [\text{ctrl}; \text{plant}](\text{req})\\
\text{init} \equiv\ &\forall[-\tau]\,(D - m \le x_n[s] - x_{n+1}[s] \le D + m\\
&\wedge V - l \le x'_{n+1}[s] \le V + l\\
&\wedge v[s] = v_0)\\
\text{ctrl} \equiv\ &\alpha := 1\\
\text{plant} \equiv\ &v'_{n+1} = \alpha(x'_n[-\tau] - x'_{n+1}[-\tau]), x'_{n+1} = v_{n+1}, x'_n = v\\
\text{req} \equiv\ &d \ge \delta
\end{aligned}$$

6.2 Network Induced Delay in Control Loops

A *Networked Control Systems* (NCS) is a feedback control system with sensing and control data transmission over a network. Sensors sample the state of the *plant* periodically and send their output as packet to an event-driven *controller*, which calculates a control signal as soon as the sensor data arrives. This signal is then transmitted to event-driven actuators in the plant, which perform action immediately on reception of the command.

The plant is considered to be continuous in time by its physical nature, whereas the controller modeled as discrete in time.

This scenario has some possible issues, such as network induced *delay* and *loss* of network packets. Due to different retardation times on the plant-controller and controller-plant connections, the plant output and the controller input may not be delivered at the same time. For this reason, the controller might not yet have received all the plant updates when it has to perform its control computations. This makes NCSs different to conventional sampled-data systems.

The plant (physical component) is modeled as time continuous and the evolution of its state $x \in \mathbb{R}^n$ by

$$\begin{cases} x'(t) &= Ax(t) + Bu(t)\\ y(t) &= Cx(t) \end{cases}$$

which depends additionally on a control signal $u \in \mathbb{R}^m$ provided by the discrete controller

$$u(kh) = -Kx(kh) \quad k \in \mathbb{N}_0$$

or

$$u(kh) = -Ky(kh) \quad k \in \mathbb{N}_0$$

if not the full state is known to the controller, but only some plant output (sensor data) $y \in \mathbb{R}^p$. The matrices A, B, K are chosen with suitable dimensions.

The network induces delays in the control loop, namely τ_{sc} between sensor and controller, as well as τ_{ca} between controller and actuator. Without loss of generality can the processing time of the controller be added to either delay.

$$\begin{cases} x'(t) = Ax(t) + Bu(t - \tau_{ca,k}) & t \in [kh + \tau_k, (k+1)h + \tau_{k+1}] \\ y(t) = Cx(t) & \\ u(t^+) = -Kx(t - \tau_{sc,k}) & t^+ \in \{kh + \tau_k \ : \ k \in \mathbb{N}_0\} \end{cases}$$

In case of a time-invariant controller, the partial delays can be combined together into a single $\tau = \tau_{sc} + \tau_{ca} + t_{\text{ctrl}}$ with the processing time of the controller.

Assuming that the delay τ_k of each sample k is less than the sampling period h and that each data sample $x(t)$ fits into a single packet, the system equations can be written as

$$\begin{cases} x'(t) = Ax(t) + Bu(t) & t \in [kh + \tau_k, (k+1)h + \tau_{k+1}] \\ y(t) = Cx(t) & \\ u(t^+) = -Kx(t - \tau_k) & t^+ \in \{kh + \tau_k \ : \ k \in \mathbb{N}_0\} \end{cases}$$

with the piecewise constant control signal $u(t^+)$ in the actuator.

This plant system can be solved using the standard *variation of constants* method for ordinary differential equations in order to express the new state variables $u(kh)$ and $x((k+1)h)$ as function of their values at the previous sampling instant.

The NCS models above can be written as hybrid systems with delay. The former is given in Program 6.2.

This example only admits a delay in the control-output u, but not in the dependence on the plant-state, what means that the differential equation describing the plant is actually of ordinary type. Thus, this example can also be written down in $\mathsf{d}\mathcal{L}$, by splitting the ODE at $t = \tau_{sc} + \tau_{ca} < h$ and storing the value $x(t)$ for later use. However, the extended syntax of $\mathsf{dd}\mathcal{L}$ allows a more natural notation.

Model 6.2 Simple NCS model with network induced delay

$$\begin{aligned} &\text{init} \to [(\text{ctrl}; \text{plant})^*](\text{req}) \\ \text{init} &\equiv h > 0 \land t = h \land T = \land\forall[-T)\, x = \land\forall[-T)\, u = \\ \text{ctrl} &\equiv (?t = h); t := 0, u := -Kx[-\tau_{sc}] \\ \text{plant} &\equiv (?t < h); x' = Ax + Bu[-\tau_{ca}], t' = 1 \,\&\, t \le h \\ \text{req} &\equiv \end{aligned}$$

7 Conclusion

We have presented a combination of first-order and modal logic for the specification and verification of safety and liveness properties of time-delay hybrid systems. This logic is essentially an extension of differential dynamic logic ($\mathsf{d}\mathcal{L}$). The approach taken in this work has been available for ODEs for several years, but has not yet been applied to DDEs. To the knowledge of the author, no work combining logics and delay differential equations has been published so far.

7.1 Related Work

Verification theory for hybrid systems governed by ordinary differential equations is well established, as can be seen by the exhaustive list of literature on $\mathsf{d}\mathcal{L}$ given in the introduction. Literature on formal methods applied to DDEs however, is rare.

Huang et al. [HFM16] describe an algorithm allowing bounded invariant verification for dynamical systems with delayed interconnections. This is done by numerically computing bounds on the trajectories of a (non-linear) delay differential equation, given a set of possible initial states (sensitivity of the solution on initial data).

Zou et al. [Zou+15] have presented an iteration method, combining numerical integration of DDEs with multiple constant delays based on interval Taylor forms with real arithmetic constraint solving. It automatically analyzes the integration-operator, which yields the Taylor coefficients for the next temporal segment, given by the method of step. This procedure allows to compute safety and stability certificates in terms of set constraints.

7.2 Future Work

There are several avenues for future work. Practical applicability could be improved by combining axioms to rules for a sequent proof calculus, as it is available for $\mathsf{d}\mathcal{L}$.

Next, we would like to include $\mathsf{dd}\mathcal{L}$ into the interactive theorem prover KeYmaera X. This would require an extension of the presented theory into the *uniform substitution calculus* of [Pla15a]. In contrast to the axiomatization presented in Chapter 5.1, this calculus uses a finite number of axioms (instead of axiom schemata) and an additional proof rule to substitute a formula by a predicate symbol, preserving the soundness of the axiom. This approach has the advantage that it is much easier to implement in a software tool. The first step towards this calculus is already done with the definition of *free, bound* and *mustbound variables* in Section 4.3.

The actual implementation of $\mathsf{dd}\mathcal{L}$ in KeYmaera X would benefit from its modularity and abstraction, since most existing axioms and lemmata could be taken from $\mathsf{d}\mathcal{L}$.

The axiom of steps ([⇥]) translates a delay differential equation into a ODE inside a loop. This indicates, that dd$\mathcal{L}$ is actually *reducible* to classical d$\mathcal{L}$. We leave the details, a formal proof and its significance for *completeness* as future work.

Differential invariants provide a powerful tool for the verification of differential equations, if one has such an invariant formula for the given equation to hand. However, finding such is in general not trivial. In the case of ordinary differential equations, different methods to automatically compute differential invariants have been presented [Pla12a]. Their adaption to DDEs would mean a tremendous progress towards practical application and would help to demonstrate the utility of dd$\mathcal{L}$ by proving more complex real-world examples.

Moreover, the class of feasible delay differential equations can be extended to include e.g. state dependent delays, functional right hand sides or uncertain delays. This would allow even more realistic and powerful models of problems, but increases the complexity of the possible dynamics to a large extent. It is not sure how far a logical approach can still capture these effects.

Bibliography

[Bim14] Katalin Bimbó. *Proof Theory – Sequent Calculi and Related Formalisms.* Discrete Mathematics and Its Applications. Boca Raton: CRC Press, 2014. ISBN: 978-1-4665-6466-4.

[Bus98] Samuel R. Buss. *An Introduction to Proof Theory.* Ed. by Samuel R. Buss. Vol. 137. Studies in Logic and the Foundations of Mathematics. Amsterdam: Elsevier, 1998. Chap. I, pp. 1–78. ISBN: 0-444-89840-9.

[BZ13] Alfredo Bellen and Marino Zennaro. *Numerical Methods for Delay Differential Equations.* Oxford: Oxford University Press, 2013. ISBN: 978-0-198-50654-6. DOI: `10.1093/acprof:oso/9780198506546.001.0001`.

[Dad02] E. Ait Dads. "Some General Results and Remarks on Delay Differential Equations". In: *Delay Differential Equations and Applications.* Ed. by Ovide Arino, Moulay Lhassan Hbid, and E. Ait Dads. Vol. 205. Proceedings of the NATO Advanced Study Institute. Marrakech, Morocco: Springer, Sept. 2002. Chap. 2, pp. 31–40. ISBN: 978-1-4020-3645-3. DOI: `10.1007/1-4020-3647-7`.

[Ern09] Thomas Erneux. *Applied Delay Differential Equations.* Vol. 3. Surveys and Tutorials in the Applied Mathematical Sciences. New York: Springer, 2009. ISBN: 978-0-387-74371-4. DOI: `10.1007/978-0-387-74372-1`.

[Fal06] Clement E. Falbo. *Some Elementary Methods for Solving Functional Differential Equations.* `http://www.mathfile.net/hicstat_fde.pdf`. Sonoma State University. 2006.

[FP16] Nathan Fulton and André Platzer. "A Logic of Proofs for Differential Dynamic Logic: Toward Independently Checkable Proof Certificates for Dynamic Logics". In: *Proceedings of the 2016 Conference on Certified Programs and Proofs (CPP 2016).* Ed. by Jeremy Avigad and Adam Chlipala. St. Petersburg, FL, USA: ACM, Jan. 2016, pp. 110–121. DOI: `10.1145/2854065.2854078`.

[Gat12] Andreas Gathmann. *Grundlagen der Mathematik.* `http://www.mathematik.uni-kl.de/agag/mitglieder/professoren/gathmann/notes/gdm/`. Class Notes TU Kaiserslautern. 2012.

[Gho+] Khalil Ghorbal et al. "Hybrid Theorem Proving of Aerospace Systems: Applications and Challenges". In: *Journal of Aerospace Information Systems* 11 (), pp. 702–713. DOI: `10.2514/1.I010178`.

[GN03] Keqin Gu and Silviu-Iulian Niculescu. "Survey on Recent Results in the Stability and Control of Time-Delay Systems". In: *Journal of Dynamic Systems, Measurement, and Control* 125.2 (2003), pp. 158–165. DOI: 10.1115/1.1569950. URL: https://www.researchgate.net/profile/Keqin_Gu/publication/225075485_Survey_on_recent_results_in_the_stability_and_control_of_time-delay_systems/links/02e7e529a977c83c4b000000.pdf.

[HFM16] Zhenqi Huang, Chuchu Fan, and Sayan Mitra. "Bounded Invariant Verification for Time-delayed Nonlinear Networked Dynamical Systems". In: *Nonlinear Analysis: Hybrid Systems* (2016). ISSN: 1751-570X. DOI: 10.1016/j.nahs.2016.05.005. URL: http://www.sciencedirect.com/science/article/pii/S1751570X16300279.

[Hod01] Wilfrid Hodges. "Classical Logic I: First-Order Logic". In: *The Blackwell Guide to Philosophical Logic.* Ed. by Lou Goble. Malden/Oxford: Blackwell Publishers, 2001. Chap. 1, pp. 9–32. ISBN: 0-631-20693-0. URL: https://cs.uwaterloo.ca/~david/cl/fol-hodges.

[HR04] Michael Huth and Mark Ryan. *Logic in Computer Science – Modelling and Reasoning About Systems.* 2. Cambridge University Press, 2004. ISBN: 0-521-54310-X.

[Jea+15a] Jean-Baptiste Jeannin et al. "A Formally Verified Hybrid System for the Next-generation Airborne Collision Avoidance System". In: *Proceedings of 21st International Conference on Tools and Algorithms for the Construction and Analysis of Systems, TACAS 2015.* Ed. by Christel Baier and Cesare Tinelli. Vol. 9035. LNCS. London, UK: Springer, Apr. 2015, pp. 21–36. DOI: 10.1007/978-3-662-46680-3_2.

[Jea+15b] Jean-Baptiste Jeannin et al. "Formal Verification of ACAS X, an Industrial Airborne Collision Avoidance System". In: *International Conference on Embedded Software, EMSOFT 2015.* Ed. by Alain Girault and Nan Guan. Amsterdam, Netherlands: IEEE Press, Oct. 2015, pp. 127–136. DOI: 10.1109/EMSOFT.2015.7318268.

[JJ98] Giorgi Japaridze and Dick de Jongh. "The Logic of Provability". In: *Handbook of Proof Theory.* Ed. by Samuel R. Buss. Vol. 137. Studies in Logic and the Foundations of Mathematics. Amsterdam: Elsevier, 1998. Chap. VII, pp. 476–546. ISBN: 0-444-89840-9.

[Kou+13] Yanni Kouskoulas et al. "Certifying the Safe Design of a Virtual Fixture Control Algorithm for a Surgical Robot". In: *Hybrid Systems: Computation and Control, HSCC'13.* Ed. by Calin Belta and Franjo Ivancic. Philadelphia, PA, USA: ACM, Apr. 2013, pp. 263–272. DOI: 10.1145/2461328.2461369.

[Loo+13] Sarah M. Loos et al. “Efficiency Analysis of Formally Verified Adaptive Cruise Controllers”. In: *Proceedings of 16th International IEEE Conference on Intelligent Transportation Systems, ITSC.* Ed. by Andreas Hegyi and Bart De Schutter. The Hague, Netherlands, Oct. 2013, pp. 1565–1570. ISBN: 978-1-4799-2914-613. DOI: `10.1109/ITSC.2013.6728453`.

[LP11] Sarah M. Loos and André Platzer. “Safe Intersections: At the Crossing of Hybrid Systems and Verification”. In: *Proceedings of 14th International IEEE Conference on Intelligent Transportation Systems (ITSC).* Ed. by Kyongsu Yi. Washington, DC, USA, Oct. 2011, pp. 1181–1186. DOI: `10.1109/ITSC.2011.6083138`.

[LPN11] Sarah M. Loos, André Platzer, and Ligia Nistor. “Adaptive Cruise Control: Hybrid, Distributed, and Now Formally Verified”. In: *Formal Methods, Proceedings of 17th International Symposium on Formal Methods, FM 2011.* Ed. by Michael Butler and Wolfram Schulte. Vol. 6664. LNCS. Limerick, Ireland: Springer, June 2011, pp. 42–56. DOI: `10.1007/978-3-642-21437-0_6`.

[LRP13] Sarah M. Loos, David W. Renshaw, and André Platzer. “Formal Verification of Distributed Aircraft Controllers”. In: *Hybrid Systems: Computation and Control, HSCC’13.* Ed. by Calin Belta and Franjo Ivancic. Philadelphia, PA, USA: ACM, Apr. 2013, pp. 125–130. DOI: `10.1145/2461328.2461350`.

[Mit+16] Stefan Mitsch et al. “Formal Verification of Obstacle Avoidance and Navigation of Ground Robots”. In: *CoRR* abs/1605.00604 (2016). URL: `http://arxiv.org/abs/1605.00604`.

[MLP12] Stefan Mitsch, Sarah M. Loos, and André Platzer. “Towards formal verification of freeway traffic control”. In: *ACM/IEEE Third International Conference on Cyber-Physical Systems (ICCPS).* Ed. by Chenyang Lu. Beijing, China: IEEE, Apr. 2012, pp. 171–180. ISBN: 978-0-7695-4695-7. DOI: `10.1109/ICCPS.2012.25`.

[Pla07] André Platzer. “A Temporal Dynamic Logic for Verifying Hybrid System Invariants”. In: *Proceedings of 5th International Symposium on Logical Foundations of Computer Science, LFCS’07.* Ed. by Sergei N. Artëmov and Anil Nerode. Vol. 4514. LNCS. New York, USA: Springer, June 2007, pp. 457–471. ISBN: 978-3-540-72732-3. DOI: `10.1007/978-3-540-72734-7_32`.

[Pla08] André Platzer. “Differential Dynamic Logics: Automated Theorem Proving for Hybrid Systems”. PhD thesis. Department of Computing Science, University of Oldenburg, 2008, p. 299.

[Pla10a] André Platzer. “Differential-algebraic Dynamic Logic for Differential-algebraic Programs”. In: *Journal of Logic and Computation* 20.1 (2010), pp. 309–352. DOI: `10.1093/logcom/exn070`.

[Pla10b] André Platzer. *Logical Analysis of Hybrid Systems – Proving Theorems for Complex Dynamics.* Heidelberg: Springer, 2010. ISBN: 978-3-642-14508-7. DOI: 10.1007/978-3-642-14509-4.

[Pla10c] André Platzer. "Quantified Differential Dynamic Logic for Distributed Hybrid Systems". In: *Proceedings of 24th International Workshop in Computer Science Logic (CSL).* Ed. by Anuj Dawar and Helmut Veith. Vol. 6247. LNCS. Brno, Czech Republic: Springer, Aug. 2010, pp. 469–483. ISBN: 978-3-642-15204-7. DOI: 10.1007/978-3-642-15205-4_36.

[Pla11] André Platzer. "Stochastic Differential Dynamic Logic for Stochastic Hybrid Programs". In: *Proceedings of International Conference on Automated Deduction, CADE'11.* Ed. by Nikolaj Bjørner and Viorica Sofronie-Stokkermans. Vol. 6803. LNCS. Wrocław, Poland: Springer, 2011, pp. 431–445. DOI: 10.1007/978-3-642-22438-6_34.

[Pla12a] André Platzer. "Logics of Dynamical Systems". In: *27th Annual IEEE Symposium on Logic in Computer Science (LICS).* Dubrovnik, Croatia: IEEE, June 2012, pp. 13–24. ISBN: 978-1-4673-2263-8. DOI: 10.1109/LICS.2012.13.

[Pla12b] André Platzer. "The Complete Proof Theory of Hybrid Systems". In: *LICS.* IEEE, 2012, pp. 541–550. ISBN: 978-1-4673-2263-8. DOI: 10.1109/LICS.2012.64.

[Pla15a] André Platzer. "A Uniform Substitution Calculus for Differential Dynamic Logic". In: *CoRR* abs/1503.01981 (2015). URL: http://arxiv.org/abs/1503.01981.

[Pla15b] André Platzer. "Differential Game Logic". In: *ACM Trans. Comput. Log.* 17.1 (2015), 1:1–1:51. ISSN: 1529-3785. DOI: 10.1145/2817824.

[PQ09] André Platzer and Jan-David Quesel. "European Train Control System: A Case Study in Formal Verification". In: *Formal Methods and Software Engineering, Proceedings of 11th International Conference on Formal Engineering Methods, ICFEM 2009.* Ed. by Karin Breitman and Ana Cavalcanti. Vol. 5885. LNCS. Rio de Janeiro, Brasil: Springer, Dec. 2009, pp. 246–265. DOI: 10.1007/978-3-642-10373-5_13.

[PW10] Jan W. Prüss and Mathias Wilke. *Gewöhnliche Differentialgleichungen und dynamische Systeme.* Basel: Springer, 2010. ISBN: 978-3-0348-0002-0. DOI: 10.1007/978-3-0348-0002-0.

[Que+16] Jan-David Quesel et al. "How to Model and Prove Hybrid Systems with KeYmaera: A Tutorial on Safety". In: *STTT* 18.1 (2016), pp. 67–91. DOI: 10.1007/s10009-015-0367-0.

[Rau10] Wolfgang Rautenberg. *A Concise Introduction to Mathematical Logic.* 3. Universitext. New York: Springer, 2010. ISBN: 978-1-4419-1220-6. DOI: 10.1007/978-1-4419-1221-3.

[Rei14] Giselle Reis. "Cut-Elimination by Resolution in Intuitionistic Logic". PhD thesis. Technische Universität Wien, 2014.

[Ric03] Jean-Pierre Richard. “Time-delay Systems: An Overview of Some Recent Advances and Open Problems”. In: *Automatica* 39.10 (2003), pp. 1667–1694. DOI: `10.1016/S0005-1098(03)00167-5`.

[Rou05] Marc R. Roussel. *Nonlinear Dynamics – Delay-differential equations.* `http://people.uleth.ca/~roussel/nld/delay.pdf`. Class Notes University of Lethbridge. 2005.

[Rud76] Walter Rudin. *Principles of Mathematical Analysis.* 3. International Series in Pure & Applied Mathematics. New York: McGraw-Hill, 1976. ISBN: 0-07-054235-X.

[Smi10] Hal Smith. *An Introduction to Delay Differential Equations with Applications to the Life Sciences.* Texts in Applied Mathematics. New York: Springer, 2010. ISBN: 978-1-4419-7646-8. DOI: `10.1007/978-1-4419-7646-8`.

[Szc14] Robert Szczelina. “Rigorous Integration of Delay Differential Equations”. PhD thesis. Jagiellonian University Krakow, 2014.

[TS00] A. S. Troelstra and H. Schwichtenberg. *Basic Proof Theory (2Nd Ed.)* New York, NY, USA: Cambridge University Press, 2000. ISBN: 0-521-77911-1.

[ZBP01] Wei Zhang, Michael S. Branicky, and Stephen M. Phillips. “Stability of Networked Control Systems”. In: *IEEE Control Systems* 21.1 (2001), pp. 84–99. ISSN: 1066-033X. DOI: `10.1109/37.898794`.

[Zou+15] Liang Zou et al. “Automatic Verification of Stability and Safety for Delay Differential Equations”. In: *Proceedings of the 27th International Conference on Computer Aided Verification (CAV 2015).* San Francisco, CA, USA: Springer, July 2015, pp. 338–355. DOI: `10.1007/978-3-319-21668-3_20`.