EPiC Series in Computer Science

Volume 33, 2015, Pages 17–30

ARQNL 2014. Automated Reasoning in Quantified Non-Classical Logics

A Logic for Verifying Metric Temporal Properties in Distributed Hybrid Systems

Ping Hou[1] and Yifei Chen[2*]

[1] Fondazione Bruno Kessler (FBK-irst), Trento, Italy
[2] School of Information Science, Nanjing Audit University, Nanjing, China
phou@cs.cmu.edu, yifeichen91@nau.edu.cn

Abstract

We introduce a logic for specifying and verifying metric temporal properties of distributed hybrid systems that combines quantified differential dynamic logic (QdℒC) for reasoning about the possible behavior of distributed hybrid systems with metric temporal logic (MTL) for reasoning about the metric temporal behavior during their operation. For our combined logic, we generalize the semantics of dynamic modalities to refer to hybrid traces instead of final states. Further, we prove that this gives a conservative extension of QdℒC for distributed hybrid systems. On this basis, we provide a modular verification calculus that reduces correctness of metric temporal behavior of distributed hybrid systems to generic temporal reasoning and then non-temporal reasoning, and prove that we obtain a complete axiomatization relative to the non-temporal base logic QdℒC.

1 Introduction

Functional specifications for distributed hybrid systems [6, 17, 22, 23] with discrete, continuous, structural, and dimension-changing dynamics usually involve a number of critical properties such as timing requirements, stability and bounded response. Metric Temporal Logic (MTL) [16] is a popular formalism for expressing such properties. The problem of verifying MTL specifications is undecidable for hybrid systems and distributed hybrid systems. Consequently, the *bounded-time* verification or falsification of such properties has been studied [10, 11, 18, 20, 27]. In addition to having successful uses in simulation-based methods, metric temporal logic has been used in deductive approaches to prove validity of formulas in calculi [12, 19]. Valid MTL formulas, however, cannot generally characterize the operations of a specific system.

Very recently, a dynamic logic, called *quantified differential dynamic logic* (QdℒC) has been introduced as a successful tool for deductively verifying distributed hybrid systems [22, 23]. QdℒC can analyze the behavior of actual distributed hybrid system models, which are specified operationally. Yet, operational distributed hybrid system models are *internalized* within

*The work of this author is supported by the National Natural Science Foundation of China (No.61202135) and the Natural Science Foundation of Jiangsu Province, China (No.BK2012472).

QdℒL formulas. However, QdℒL only considers the behavior of distributed hybrid systems at final states, which is insufficient for verifying properties that hold within certain time bounds, throughout the execution of the system.

We close this gap of expressivity by combining QdℒL with metric temporal logic. In this paper, we introduce a logic, called *quantified differential metric temporal dynamic logic* (QdMTL), which provides modalities for quantifying over traces of distributed hybrid systems based on QdℒL. We equip QdMTL with metric temporal operators to state what is true all along a time interval of a trace or at some time point during a trace. In this paper, we modify the semantics of the dynamic modality $[\alpha]$ to refer to all *traces* of α instead of all final states reachable with α (similarly for $\langle\alpha\rangle$). For instance, the formula $[\alpha]\Box_{\mathcal{I}}\,\phi$ expresses that ϕ is true at each state within the time interval $\mathcal{I}$ along all traces of the distributed hybrid system α. With this, QdMTL can also be used to verify temporal statements about the behavior of α at intermediate states during system runs. As in our non-temporal dynamic logic QdℒL, we use *quantified hybrid programs* as an operational model for distributed hybrid systems, since they admit a uniform compositional treatment of interacting discrete transitions, continuous evolutions, and structural/dimension changes in logic.

As a semantical foundation for combined quantified temporal dynamic formulas, we introduce a hybrid trace semantics for QdMTL. We prove that QdMTL is a conservative extension of QdℒL: for non-temporal specifications, trace semantics is equivalent to the non-temporal transition semantics of QdℒL [22, 23].

As a means for verification, we introduce a sequent calculus for QdMTL that successively reduces temporal statements about traces of quantified hybrid programs to non-temporal QdℒL formulas. In this way, we make the intuition formally precise that metric temporal properties can be checked by augmenting proofs with appropriate assertions about intermediate states. Like in [22, 23], our calculus works compositionally. It decomposes correctness statements about quantified hybrid programs structurally into corresponding statements about its parts by symbolic transformation.

Our approach combines the advantages of QdℒL in reasoning about the behaviour of operational distributed hybrid system models with those of metric temporal logic to verify temporal statements about traces. Our first contribution is the logic QdMTL, which provides a coherent foundation for reasoning about the metric temporal behaviour of operational models of distributed hybrid systems. The main contribution is our calculus for deductively verifying temporal statements about distributed hybrid systems, which is a complete axiomatization relative to non-temporal QdℒL.

2 Related Work

Multi-party distributed control has been suggested for car control [15] and air traffic control [7]. Due to limits in verification technology, no formal analysis of metric temporal statements about the distributed hybrid dynamics has been possible for these systems yet. Analysis results include discrete message handling [15] or collision avoidance for two participants [7].

The importance of understanding dynamic/reconfigurable distributed hybrid systems was recognized in modeling languages SHIFT [6] and R-Charon [17]. They focused on simulation/-compilation [6] or the development of a semantics [17], so that no verification is possible yet.

A variety of simulation-based approaches, like [10, 11, 18, 20, 27], have been proposed for the verification or falsification of metric temporal logic properties in hybrid systems, which, however, cannot be used to verify temporal properties in distributed hybrid systems.

Our approach is completely different. It is based on first-order structures and dynamic logic. We focus on developing a logic that supports metric temporal, generic temporal, and non-temporal statements about distributed hybrid dynamics and is amenable to automated theorem proving in the logic itself.

Based on [25], Beckert and Schlager [3] added separate trace modalities to dynamic logic and presented a relatively complete calculus. Their approach only handles discrete state spaces and conventional temporal modalities. In contrast, QdMTL works for hybrid programs with continuous and structural/dimensional dynamics and metric temporal quantifiers.

Davoren and Nerode [5] extended the propositional modal μ-calculus with a semantics in hybrid systems and examine topological aspects. In [4], Davoren et al. gave a semantics in terms of general flow systems for a generalisation of CTL* [9]. In both cases, the authors of [5] and [4] provided Hilbert-style calculi to prove formulas that are valid for all systems simultaneously using abstract actions.

The strength of our logic primarily is that it is a first-order quantified dynamic logic: it handles actual quantified hybrid programs rather than only abstract actions of unknown effect. Our calculus directly supports verification of quantified hybrid programs with first-order definable flows and structures.

3 Syntax of Quantified Differential Metric Temporal Dynamic Logic

As a formal logic for verifying metric temporal specifications of distributed hybrid systems, we introduce *quantified differential metric temporal dynamic logic* (QdMTL). QdMTL extends dynamic logic for reasoning about system runs [13] with many-sorted first-order logic for reasoning about all ($\forall i : A\ \phi$) or some ($\exists i : A\ \phi$) objects of a sort A, e.g., the sort of all aircraft, and three other concepts:

Quantified hybrid programs. The behavior of distributed hybrid systems can be described by quantified hybrid programs [22, 23], which generalize regular programs from dynamic logic [13] to distributed hybrid changes. The distinguishing feature of quantified hybrid programs is that they provide uniform discrete transitions, continuous evolutions, and structural/dimension changes along quantified assignments and quantified differential equations, which can be combined by regular control operations.

Modal operators. Modalities of dynamic logic express statements about all possible behavior ($[\alpha]\pi$) of a system α, or about the existence of a trace ($\langle\alpha\rangle\pi$), satisfying condition π. Unlike in standard dynamic logic, α is a model of a distributed hybrid system. We use quantified hybrid programs to describe α as in [22, 23]. Yet, unlike in standard dynamic logic [13] or quantified differential dynamic logic (Qd$\mathcal{L}$) [22, 23], π is a *trace formula* in QdMTL, and π can refer to every state that occur *during* an arbitrary time interval of a trace using metric temporal operators.

Metric temporal operators. For QdMTL, the metric temporal trace formula $\Box_{\mathcal{I}}\,\phi$ expresses that the formula ϕ holds all along the time interval $\mathcal{I}$ of a trace selected by $[\alpha]$ or $\langle\alpha\rangle$. For instance, the state formula $\langle\alpha\rangle\Box_{\mathcal{I}}\,\phi$ says that the state formula ϕ holds at every state within the time interval $\mathcal{I}$ of at least one trace of α. Dually, the trace formula $\Diamond_{\mathcal{I}}\,\phi$ expresses that ϕ holds at some point during the time interval $\mathcal{I}$ of such a trace . It can occur in a state formula $\langle\alpha\rangle\Diamond_{\mathcal{I}}\,\phi$ to express that there is such a state in the time interval $\mathcal{I}$ of some trace of α, or as $[\alpha]\Diamond_{\mathcal{I}}\,\phi$ to say that along each trace there is a state within the time interval $\mathcal{I}$ satisfying ϕ. In this paper,

the primary focus of attention is on homogeneous combination of path and trace quantifiers like $[\alpha]\Box_{\mathcal{I}}\,\phi$ or $\langle\alpha\rangle\Diamond_{\mathcal{I}}\,\phi$.

3.1 Quantified Hybrid Programs

We let $\mathbb{R}$ (resp. $\mathbb{R}_{\geq 0}$, $\mathbb{N}$) denote the set of reals (resp. non-negative reals, non-negative integers). An *interval* of $\mathbb{R}$ (resp. $\mathbb{R}_{\geq 0}$, $\mathbb{N}$) is a convex subset of $\mathbb{R}$ (resp. $\mathbb{R}_{\geq 0}$, $\mathbb{N}$). QdMTL supports a (finite) number of object *sorts*, e.g., the sort of all aircraft, or the sort of all cars. For continuous quantities of distributed hybrid systems like positions or velocities, we add the sort $\mathbb{R}$ for real numbers. *Terms* of QdMTL are built from a set of (sorted) function/variable symbols as in many-sorted first-order logic. For representing appearance and disappearance of objects while running quantified hybrid programs, we use an existence function symbol $\mathsf{E}(\cdot)$ that has value $\mathsf{E}(o) = 1$ if object o exists, and has value $\mathsf{E}(o) = 0$ when object o disappears or has not been created yet. We use 0, 1, $+$, $-$, $\cdot$, /with the usual notation and fixed semantics for real arithmetic. For $n \geq 0$ we abbreviate $f(s_1, \ldots, s_n)$ by $f(\boldsymbol{s})$ using vectorial notation and we use $\boldsymbol{s} = \boldsymbol{t}$ for element-wise equality.

As a system model for distributed hybrid systems, QdMTL uses *quantified hybrid programs* (QHP) [22, 23]. The quantified hybrid programs occurring in dynamic modalities of QdMTL are regular programs from dynamic logic [13] to which quantified assignments and quantified differential equation systems for distributed hybrid dynamics are added. QHPs are defined by the following grammar (α, β are QHPs, θ a term, i a variable of sort A, f is a function symbol, $\boldsymbol{s}$ is a vector of terms with sorts compatible to f, and χ is a formula of first-order logic):

$$\alpha, \beta ::= \forall i : A\ f(\boldsymbol{s}) := \theta \mid \forall i : A\ f(\boldsymbol{s})' = \theta\ \&\ \chi \mid ?\chi \mid \alpha \cup \beta \mid \alpha; \beta \mid \alpha^*$$

The effect of *quantified assignment* $\forall i : A\ f(\boldsymbol{s}) := \theta$ is an instantaneous discrete jump assigning θ to $f(\boldsymbol{s})$ simultaneously for all objects i of sort A. The quantified hybrid program $\forall i : C\ a(i) := a(i) + 1$, for example, expresses that all cars i of sort C simultaneously increase their acceleration $a(i)$. The effect of *quantified differential equation* $\forall i : A\ f(\boldsymbol{s})' = \theta\ \&\ \chi$ is a continuous evolution where, for all objects i of sort A, all differential equations $f(\boldsymbol{s})' = \theta$ hold and formula χ holds throughout the evolution (the state remains in the region described by χ). The dynamics of QHPs changes the interpretation of terms over time: $f(\boldsymbol{s})'$ is intended to denote the derivative of the interpretation of the term $f(\boldsymbol{s})$ over time during continuous evolution, not the derivative of $f(\boldsymbol{s})$ by its argument $\boldsymbol{s}$. For $f(\boldsymbol{s})'$ to be defined, we assume f is an $\mathbb{R}$-valued function symbol. For simplicity, we assume that f does not occur in $\boldsymbol{s}$. In most quantified assignments/differential equations $\boldsymbol{s}$ is just i. For instance, the following QHP expresses that all cars i of sort C drive by $\forall i : C\ x(i)'' = a(i)$ such that their position $x(i)$ changes continuously according to their respective acceleration $a(i)$. Time itself is implicit. If a clock variable c is needed in a QHP, it can be axiomatized by $c' = 1$, which is equivalent to $\forall i : A\ c' = 1$ where i does not occur in c. For such *vacuous quantification* (i does not occur anywhere), we may omit $\forall i : A$ from assignments and differential equations.

The effect of test $?\chi$ is a *skip* (i.e., no change) if formula χ is true in the current state and *abort* (blocking the system run by a failed assertion), otherwise. *Nondeterministic choice* $\alpha \cup \beta$ is for alternatives in the behavior of the distributed hybrid system. In the *sequential composition* $\alpha; \beta$, QHP β starts after α finishes (β never starts if α continues indefinitely). *Nondeterministic repetition* α^* repeats α an arbitrary number of times, possibly zero times.

Structural dynamics of distributed hybrid systems corresponds to quantified assignments to function terms and we model the appearance of new participants in the distributed hybrid system, e.g., new aircraft appearing into the local flight scenario, by a program $n := \mathsf{new}\ A$ (see [22, 23] for details).

3.2 State and Trace Formulas

The formulas of QdMTL are defined similarly to first-order dynamic logic plus many-sorted first-order logic. However, the modalities $[\alpha]$ and $\langle\alpha\rangle$ accept trace formulas that refer to the temporal behavior of *all* states within a time interval $\mathcal{I}$ along a trace. Inspired by CTL and CTL* [8, 9], we distinguish between state formulas, which are true or false in states, and trace formulas, which are true or false for system traces.

The *state formulas* of QdMTL are defined by the following grammar (ϕ, ψ are state formulas, π is a trace formula, θ_1, θ_2 are terms of the same sort, i is a variable of sort A, and α is a QHP):

$$\phi, \psi ::= \theta_1 = \theta_2 \mid \theta_1 \geq \theta_2 \mid \neg\phi \mid \phi \wedge \psi \mid \forall i : A\ \phi \mid \exists i : A\ \phi \mid [\alpha]\pi \mid \langle\alpha\rangle\pi$$

We use standard abbreviations to define $\leq, >, <, \vee, \rightarrow$. Sorts $A \neq \mathbb{R}$ have no ordering and only $\theta_1 = \theta_2$ is allowed. For sort $\mathbb{R}$, we abbreviate $\forall x : \mathbb{R}\ \phi$ by $\forall x \phi$.

The *trace formulas* of QdMTL are defined by the following grammar (π is a trace formula, ϕ is a state formula, and $\mathcal{I}$ is an interval of $\mathbb{R}_{\geq 0}$):

$$\pi ::= \phi \mid \Box_{\mathcal{I}}\ \phi \mid \Diamond_{\mathcal{I}}\ \phi$$

We omit the time constraint on temporal operators $\Box$ and $\Diamond$ when $[0, +\infty)$ is assumed. Thus the metric temporal operators $\Box$ and $\Diamond$ coincide with the conventional *always* and *eventually* temporal operators of CTL.

Formulas with only conventional temporal operators $\Box$ and $\Diamond$, i.e., metric temporal operators $\Box_{[0,+\infty)}$ and $\Diamond_{[0,+\infty)}$, are called *non-metric temporal* formulas. Formulas without metric temporal operators $\Box_{\mathcal{I}}$ and $\Diamond_{\mathcal{I}}$ are called *non-temporal* QdL *formulas* [22, 23]. Unlike in CTL, state formulas are true on a trace if they hold for the *last* state of a trace, not for the first. Thus, $[\alpha]\phi$ expresses that ϕ is true at the end of each trace of α. In contrast, $[\alpha]\Box_{\mathcal{I}}\ \phi$ expresses that ϕ is true all along all states within $\mathcal{I}$ of every trace of α. This combination gives a smooth embedding of non-temporal QdL into QdMTL and makes it possible to define a compositional calculus. Like CTL, QdMTL allows nesting with a branching time semantics [8], e.g., $[\alpha]\Box_{[0,4)}\ ((\forall i : C\ x(i) \geq 5) \rightarrow \langle\beta\rangle\Diamond_{(2,3]}\ (\forall i : C\ x(i) \leq 1))$. In the following, all formulas and terms have to be well-typed. For short notation, we allow conditional terms of the form $\mathsf{if}\ \phi\ \mathsf{then}\ \theta_1\ \mathsf{else}\ \theta_2\ \mathsf{fi}$ (where θ_1 and θ_2 have the same sort). This term evaluates to θ_1 if the formula ϕ is true and to θ_2 otherwise. We consider formulas with conditional terms as abbreviations, e.g., $\psi(\mathsf{if}\ \phi\ \mathsf{then}\ \theta_1\ \mathsf{else}\ \theta_2)$ for $(\phi \rightarrow \psi(\theta_1)) \wedge (\neg\phi \rightarrow \psi(\theta_2))$.

Example 1. *Let C be the sort of all cars. By $x(i)$, we denote the position of car i, by $v(i)$ its velocity and by $a(i)$ its acceleration. Then the* QdMTL *formula*

$$(\forall i : C\ x(i) \geq 0) \rightarrow [\forall i : C\ x(i)' = v(i), v(i)' = a(i)\ \&\ v(i) \geq 0]\Box_{[0,2]}\ (\forall i : C\ x(i) \geq 0)$$

says that, if all cars start at a point to the right of the origin and we only allow them to evolve as long as all of them have nonnegative velocity, then they will always *stay up to the right of the origin within 2 time units. In this case, the QHP just consists of a quantified differential equation expressing that the position $x(i)$ of car i evolves over time according to the velocity $v(i)$, which evolves according to its acceleration $a(i)$. The constraint $v(i) \geq 0$ expresses that the cars never move backwards, which otherwise would happen eventually in the case of braking $a(i) < 0$. This formula is indeed valid, and we would be able to use the techniques developed in this paper to prove it.*

4 Semantics of Quantified Differential Metric Temporal Dynamic Logic

In standard dynamic logic [13] and the logic $\mathsf{Qd}\mathcal{L}$ [22, 23], modalities only refer to the final states of system runs and the semantics is a reachability relation on states: State τ is reachable from state σ using α if there is a run of α which terminates in τ when started in σ. For QdMTL, however, formulas can refer to intermediate states in any time interval along runs as well. Thus, the behavior of distributed hybrid systems is modeled in terms of *hybrid traces*. Hybrid traces describe the evolution of system variables in every point of time. Such evolution is allowed to have discontinuous points corresponding to changes caused by discrete jumps in state space and continuous phases governed by different differential equations with different slopes.

4.1 Trace Semantics of Quantified Hybrid Programs

A *state* σ associates an infinite set $\sigma(A)$ of objects with each sort A, and it associates a function $\sigma(f)$ of appropriate type with each function symbol f, including $\mathsf{E}(\cdot)$. For simplicity, σ also associates a value $\sigma(i)$ of appropriate type with each variable i. The interpretation of $0, 1, +, -, \cdot, /$ is that of real arithmetic. We assume constant domain for each sort A: all states σ, τ share the same infinite domains $\sigma(A) = \tau(A)$. Sorts $A \neq C$ are disjoint: $\sigma(A) \cap \sigma(C) = \emptyset$. The set of all states is denoted by $\mathcal{S}$. The state σ_i^e agrees with σ except for the interpretation of variable i, which is changed to e. In addition, we distinguish a state Λ to denote the failure of a system run when it is *aborted* due to a test $?\chi$ that yields *false*. In particular, Λ can only occur at the end of an aborted system run and marks that there is no further extension. Given an interval I of $\mathbb{R}$ and a real number t, we write $I + t$ and $I - t$ for the intervals $\{t' \in \mathbb{R} \mid t' - t \in I\}$ and $\{t' \in \mathbb{R} \mid t' + t \in I\}$, respectively, and we denote with $le(I)$ and $ue(I)$ the lower and upper endpoints of I, respectively. Given an interval $I = [t, t']$ (resp. $I = (t, t']$, $I = [t, t')$, $I = (t, t')$, $I = [t, +\infty)$, $I = (t, +\infty)$) of $\mathbb{R}$ and a variable x of sort $\mathbb{R}$, we write $x \in I$ for the formula $x \geq t \wedge x \leq t'$ (resp. $x > t \wedge x \leq t'$, $x \geq t \wedge x < t'$, $x > t \wedge x < t'$, $x \geq t$, $x > t$).

Definition 2 (Hybrid Trace)**.** *A* hybrid trace *is either an infinite sequence* $\langle \overline{\nu}, \overline{I} \rangle = (\langle \nu_0, I_0 \rangle, \langle \nu_1, I_1 \rangle, \langle \nu_2, I_2 \rangle, \ldots)$ *such that*

- $\nu_i : [0, r_i] \to \mathcal{S}$ *is a function with duration* $r_i \in \mathbb{R}_{\geq 0}$ *for all* $i \in \mathbb{N}$,
- I_i *is a closed interval of* $\mathbb{R}_{\geq 0}$ ($I_i = [t, t']$ *for some* $t, t' \in \mathbb{R}_{\geq 0}$ *with* $t \leq t'$) *for all* $i \in \mathbb{N}$,
- $le(I_i) = \sum_{k=0}^{i-1} r_k$ *for all* $i > 0$ *and* $ue(I_i) = \sum_{k=0}^{i} r_k$ *for all* $i \in \mathbb{N}$,
- $ue(I_i) = le(I_{i+1})$ *for all* $i \in \mathbb{N}$,
- *the intervals cover* $\mathbb{R}_{\geq 0}$ *: every non-negative real belongs to some interval* I_i (*thus* $le(I_0) = 0$),

or a finite sequence $\langle \overline{\nu}, \overline{I} \rangle = (\langle \nu_0, I_0 \rangle, \langle \nu_1, I_1 \rangle, \ldots, \langle \nu_n, I_n \rangle)$ *such that*

- $\nu_i : [0, r_i] \to \mathcal{S}$ *is a function with duration* $r_i \in \mathbb{R}_{\geq 0}$ *for all* $0 \leq i \leq n$,
- I_i *is an interval of* $\mathbb{R}_{\geq 0}$ *for all* $0 \leq i \leq n$,
- *for all* $0 \leq i < n$, $I_i = [t, t']$ *for some* $t, t' \in \mathbb{R}_{\geq 0}$ *with* $t \leq t'$,
- $le(I_i) = \sum_{k=0}^{i-1} r_k$ *for all* $0 < i \leq n$ *and* $ue(I_i) = \sum_{k=0}^{i} r_k$ *for all* $0 \leq i < n$,
- $ue(I_i) = le(I_{i+1})$ *for all* $0 \leq i < n$,

- *the intervals cover* $\mathbb{R}_{\geq 0}$ *: every non-negative real belongs to some interval* I_i *(thus* $le(I_0) = 0$ *and* $I_n = [le(I_n), +\infty)$*).*

Further, for a state $\sigma \in \mathcal{S}$*,* $\hat{\sigma} : 0 \mapsto \sigma$ *is the* point flow *at* σ *with duration* 0*. A hybrid trace* terminates *if it is a finite sequence* $(\langle \nu_0, I_0\rangle, \langle \nu_1, I_1\rangle, \ldots, \langle \nu_n, I_n\rangle)$ *and* $\nu_n(r_n) \neq \Lambda$*. In that case, the last state* $\nu_n(r_n)$ *is denoted by* last $\langle \overline{\nu}, \overline{I}\rangle$ *and the total duration* $\sum_{i=0}^{n} r_i$ *is denoted by* $\mathcal{D}(\langle \overline{\nu}, \overline{I}\rangle)$*. The first state* $\nu_0(0)$ *is denoted by* first $\langle \overline{\nu}, \overline{I}\rangle$*. For a hybrid trace* $\langle \overline{\nu}, \overline{I}\rangle$ *and a time point* $t \in \mathbb{R}_{\geq 0}$*, the set of states of* $\langle \overline{\nu}, \overline{I}\rangle$ *at* t*, denoted by* $\langle \overline{\nu}, \overline{I}\rangle(t)$*, is*

$$\begin{cases} \{\nu_k(\zeta) \mid t \in I_k \text{ and } le(I_k) + \zeta = t\} & \text{if } \langle \overline{\nu}, \overline{I}\rangle \text{ is a finite hybrid trace } (\langle \nu_0, I_0\rangle, \\ & \langle \nu_1, I_1\rangle, \ldots, \langle \nu_n, I_n\rangle) \text{ and } t \leq \mathcal{D}(\langle \overline{\nu}, \overline{I}\rangle) \\ \{\text{last } \langle \overline{\nu}, \overline{I}\rangle\} & \text{if } \langle \overline{\nu}, \overline{I}\rangle \text{ is a finite hybrid trace } (\langle \nu_0, I_0\rangle, \\ & \langle \nu_1, I_1\rangle, \ldots, \langle \nu_n, I_n\rangle) \text{ and } t > \mathcal{D}(\langle \overline{\nu}, \overline{I}\rangle) \\ \{\nu_k(\zeta) \mid t \in I_k \text{ and } le(I_k) + \zeta = t\} & \text{if } \langle \overline{\nu}, \overline{I}\rangle \text{ is an infinite hybrid trace} \end{cases}$$

Unlike in [1, 14], the definition of traces also admits finite traces of bounded duration, which is necessary for compositionality of traces in $\alpha;\beta$. The semantics of quantified hybrid programs α as the set $\mu(\alpha)$ of its possible traces depends on valuations $\sigma[\![\cdot]\!]$ of formulas and terms at intermediate states σ. The valuation of formulas will be defined in Definition 4. Especially, we use $\sigma_i^e[\![\cdot]\!]$ to denote the valuations of terms and formulas in state σ_i^e, i.e., in state σ with i interpreted as e.

Definition 3 (Trace Semantics of Quantified Hybrid Programs)**.** *The* trace semantics*,* $\mu(\alpha)$*,* of a quantified hybrid program α*, is the set of all its possible hybrid traces and is defined inductively as follows:*

1. $\mu(\forall i : A\ f(\boldsymbol{s}) := \theta) = \{(\langle \hat{\sigma}, [0,0]\rangle, \langle \hat{\tau}, [0,+\infty)\rangle) : \sigma \in \mathcal{S}$ *and state* τ *is identical to* σ *except that at each position* $\boldsymbol{o}$ *of* f*: if* $\sigma_i^e[\![\boldsymbol{s}]\!] = \boldsymbol{o}$ *for some object* $e \in \sigma(A)$*, then* $\tau(f)(\sigma_i^e[\![\boldsymbol{s}]\!]) = \sigma_i^e[\![\theta]\!]$*.*$\}$
2. $\mu(\forall i : A\ f(\boldsymbol{s})' = \theta\ \&\ \chi) = \{(\langle \varphi, [0,+\infty)\rangle) : 0 \leq r \in \mathbb{R}$ *and* $\varphi : [0,r] \to \mathcal{S}$ *is a function satisfying the following conditions. At each time* $t \in [0,r]$*, state* $\varphi(t)$ *is identical to* $\varphi(0)$*, except that at each position* $\boldsymbol{o}$ *of* f *: if* $\sigma_i^e[\![\boldsymbol{s}]\!] = \boldsymbol{o}$ *for some object* $e \in \sigma(A)$*, then, at each time* $\zeta \in [0,r]$*:*
   - *The differential equations hold and derivatives exist (trivial for* $r = 0$*) :*
   $$\frac{\mathsf{d}(\varphi(t)_i^e[\![f(\boldsymbol{s})]\!])}{\mathsf{d}t}(\zeta) = (\varphi(\zeta)_i^e[\![\theta]\!])$$
   - *The evolution domains is respected:* $\varphi(\zeta)_i^e[\![\chi]\!] = true$*.*$\}$
3. $\mu(?\chi) = \{(\langle \hat{\sigma}, [0,+\infty)\rangle) : \sigma[\![\chi]\!] = true\} \cup \{(\langle \hat{\sigma}, [0,0]\rangle, \langle \hat{\Lambda}, [0,+\infty)\rangle) : \sigma[\![\chi]\!] = false\}$
4. $\mu(\alpha \cup \beta) = \mu(\alpha) \cup \mu(\beta)$
5. $\mu(\alpha;\beta) = \{\langle \overline{\nu}, \overline{I}\rangle \circ \langle \overline{\varsigma}, \overline{J}\rangle : \langle \overline{\nu}, \overline{I}\rangle \in \mu(\alpha), \langle \overline{\varsigma}, \overline{J}\rangle \in \mu(\beta)$ *when* $\langle \overline{\nu}, \overline{I}\rangle \circ \langle \overline{\varsigma}, \overline{J}\rangle$ *is*

defined}*; the composition of* $\langle \overline{\nu}, \overline{I} \rangle$ *and* $\langle \overline{\varsigma}, \overline{J} \rangle$*, denoted by* $\langle \overline{\nu}, \overline{I} \rangle \circ \langle \overline{\varsigma}, \overline{J} \rangle$*, is*

$$\begin{cases} (\langle \nu_0, I_0 \rangle, \ldots, \langle \nu_n, [\mathrm{le}(I_n), \mathcal{D}(\langle \overline{\nu}, \overline{I} \rangle)] \rangle, \\ \langle \varsigma_0, J_0 + \mathcal{D}(\langle \overline{\nu}, \overline{I} \rangle) \rangle, \ldots, \langle \varsigma_m, J_m + \mathcal{D}(\langle \overline{\nu}, \overline{I} \rangle) \rangle) & \text{if } \langle \overline{\nu}, \overline{I} \rangle = (\langle \nu_0, I_0 \rangle, \ldots, \langle \nu_n, I_n \rangle) \\ & \text{terminates at } \langle \nu_n, I_n \rangle,\ \langle \overline{\varsigma}, \overline{J} \rangle = \\ & (\langle \varsigma_0, J_0 \rangle, \ldots, \langle \varsigma_m, J_m \rangle),\ \text{and} \\ & \text{last } \langle \overline{\nu}, \overline{I} \rangle = \text{first } \langle \overline{\varsigma}, \overline{J} \rangle \\ (\langle \nu_0, I_0 \rangle, \ldots, \langle \nu_n, [\mathrm{le}(I_n), \mathcal{D}(\langle \overline{\nu}, \overline{I} \rangle)] \rangle, \\ \langle \varsigma_0, J_0 + \mathcal{D}(\langle \overline{\nu}, \overline{I} \rangle) \rangle, \langle \varsigma_1, J_1 + \mathcal{D}(\langle \overline{\nu}, \overline{I} \rangle) \rangle, \ldots) & \text{if } \langle \overline{\nu}, \overline{I} \rangle = (\langle \nu_0, I_0 \rangle, \ldots, \langle \nu_n, I_n \rangle) \\ & \text{terminates at } \langle \nu_n, I_n \rangle,\ \langle \overline{\varsigma}, \overline{J} \rangle = \\ & (\langle \varsigma_0, J_0 \rangle, \langle \varsigma_1, J_1 \rangle, \ldots),\ \text{and} \\ & \text{last } \langle \overline{\nu}, \overline{I} \rangle = \text{first } \langle \overline{\varsigma}, \overline{J} \rangle \\ \langle \overline{\nu}, \overline{I} \rangle & \text{if } \langle \overline{\nu}, \overline{I} \rangle \text{ does not terminate} \\ \text{not defined} & \text{otherwise} \end{cases}$$

6. $\mu(\alpha^*) = \bigcup_{n \in \mathbb{N}} \mu(\alpha^n)$*, where* $\alpha^{n+1} := (\alpha^n; \alpha)$ *for* $n \geq 1$*, as well as* $\alpha^1 := \alpha$ *and* $\alpha^0 := (?\mathit{true})$*.*

4.2 Valuation of State and Trace Formulas

Definition 4 (Valuation of Formulas)**.** *The valuation of state and trace formulas is defined respectively. For state formulas, the* valuation $\sigma[\![\cdot]\!]$ *with respect to state* σ *is defined as follows:*

1. $\sigma[\![\theta_1 = \theta_2]\!] = \mathit{true}$ *iff* $\sigma[\![\theta_1]\!] = \sigma[\![\theta_2]\!]$*; accordingly for* $\geq$*.*
2. $\sigma[\![\phi \wedge \psi]\!] = \mathit{true}$ *iff* $\sigma[\![\phi]\!] = \mathit{true}$ *and* $\sigma[\![\psi]\!] = \mathit{true}$*; accordingly for* $\neg$*.*
3. $\sigma[\![\forall i : A\ \phi]\!] = \mathit{true}$ *iff* $\sigma_i^e[\![\phi]\!] = \mathit{true}$ *for all objects* $e \in \sigma(A)$*.*
4. $\sigma[\![\exists i : A\ \phi]\!] = \mathit{true}$ *iff* $\sigma_i^e[\![\phi]\!] = \mathit{true}$ *for some object* $e \in \sigma(A)$*.*
5. $\sigma[\![[\alpha]\pi]\!] = \mathit{true}$ *iff for each hybrid trace* $\langle \overline{\nu}, \overline{I} \rangle \in \mu(\alpha)$ *that starts in* $\text{first } \langle \overline{\nu}, \overline{I} \rangle = \sigma$*, if* $\langle \overline{\nu}, \overline{I} \rangle[\![\pi]\!]$ *is defined, then* $\langle \overline{\nu}, \overline{I} \rangle[\![\pi]\!] = \mathit{true}$*.*
6. $\sigma[\![\langle \alpha \rangle \pi]\!] = \mathit{true}$ *iff there is a hybrid trace* $\langle \overline{\nu}, \overline{I} \rangle \in \mu(\alpha)$ *starting in* $\text{first } \langle \overline{\nu}, \overline{I} \rangle = \sigma$ *such that* $\langle \overline{\nu}, \overline{I} \rangle[\![\pi]\!]$ *is defined and* $\langle \overline{\nu}, \overline{I} \rangle[\![\pi]\!] = \mathit{true}$*.*

For trace formulas, the valuation $\langle \overline{\nu}, \overline{I} \rangle[\![\cdot]\!]$ *with respect to hybrid trace* $\langle \overline{\nu}, \overline{I} \rangle$ *is defined inductively as follows:*

1. *If* ϕ *is a state formula, then* $\langle \overline{\nu}, \overline{I} \rangle[\![\phi]\!] = \text{last } \langle \overline{\nu}, \overline{I} \rangle[\![\phi]\!]$ *if* $\langle \overline{\nu}, \overline{I} \rangle$ *terminates, whereas* $\langle \overline{\nu}, \overline{I} \rangle[\![\phi]\!]$ *is not defined if* $\langle \overline{\nu}, \overline{I} \rangle$ *does not terminate.*
2. $\langle \overline{\nu}, \overline{I} \rangle[\![\Box_{\mathcal{I}}\ \phi]\!] = \mathit{true}$ *iff* $\sigma[\![\phi]\!] = \mathit{true}$ *for every time point* $t \in \mathcal{I}$ *and every state* $\sigma \in \langle \overline{\nu}, \overline{I} \rangle(t)$ *with* $\sigma \neq \Lambda$*.*
3. $\langle \overline{\nu}, \overline{I} \rangle[\![\Diamond_{\mathcal{I}}\ \phi]\!] = \mathit{true}$ *iff* $\sigma[\![\phi]\!] = \mathit{true}$ *for some time point* $t \in \mathcal{I}$ *and some state* $\sigma \in \langle \overline{\nu}, \overline{I} \rangle(t)$ *with* $\sigma \neq \Lambda$*.*

As usual, a (state) formula is *valid* if it is true in all states. Further for (state) formula ϕ and state σ we write $\sigma \models \phi$ iff $\sigma[\![\phi]\!] = \mathit{true}$. We write $\sigma \not\models \phi$ iff $\sigma[\![\phi]\!] = \mathit{false}$. Likewise, for trace formula π and hybrid trace $\langle \overline{\nu}, \overline{I} \rangle$ we write $\langle \overline{\nu}, \overline{I} \rangle \models \pi$ iff $\langle \overline{\nu}, \overline{I} \rangle[\![\pi]\!] = \mathit{true}$ and $\langle \overline{\nu}, \overline{I} \rangle \not\models \pi$ iff $\langle \overline{\nu}, \overline{I} \rangle[\![\pi]\!] = \mathit{false}$. In particular, we only write $\langle \overline{\nu}, \overline{I} \rangle \models \pi$ or $\langle \overline{\nu}, \overline{I} \rangle \not\models \pi$ if $\langle \overline{\nu}, \overline{I} \rangle[\![\pi]\!]$ is defined, which it is not the case if π is a state formula and $\langle \overline{\nu}, \overline{I} \rangle$ does not terminate.

4.3 Conservative Temporal Extension

The following result shows that the extension of QdMTL by metric temporal operators does not change the meaning of non-temporal formulas. The trace semantics given in Definition 4 is equivalent to the final state reachability relation semantics [22, 23] for the sublogic $\mathsf{Qd}\mathcal{L}$ of QdMTL.

Proposition 5. *The logic* QdMTL *is a* conservative extension *of non-temporal* $\mathsf{Qd}\mathcal{L}$*, i.e., the set of valid* $\mathsf{Qd}\mathcal{L}$ *formulas is the same with respect to transition reachability semantics of* $\mathsf{Qd}\mathcal{L}$ *as with respect to the trace semantics of* QdMTL*.*

5 Proof Calculus for Metric Temporal Properties

In this section, we introduce a sequent calculus for verifying metric temporal specifications of distributed hybrid systems in QdMTL. With the basic idea being to perform a symbolic decomposition, the calculus transforms quantified hybrid programs successively into simpler logical formulas describing their effects. Statements about the metric temporal behavior of a quantified hybrid program are first converted to equivalent non-metric temporal statements without quantitative time constraints and then reduced to corresponding non-temporal statements about the intermediate states.

5.1 Proof Rules

In Fig. 1, we present a proof calculus for QdMTL that inherits the proof rules of $\mathsf{Qd}\mathcal{L}$ from [22, 23] and adds new proof rules for metric temporal modalities. We use the sequent notation informally for a systematic proof structure. A *sequent* is of the form $\Gamma \to \Delta$, where the *antecedent* Γ and *succedent* Δ are finite sets of formulas. The semantics of $\Gamma \to \Delta$ is that of the formula $\bigwedge_{\phi\in\Gamma}\phi \to \bigvee_{\psi\in\Delta}\psi$ and will be treated as an abbreviation. As usual in sequent calculus, the proof rules are applied backwards from the *conclusion* (goal below horizontal bar) to the *premise*s (subgoals above bar).

Definition 6. *Let α be a quantified hybrid program and c a clock variable, i,e., a variable of sort $\mathbb{R}$ changing with constant slope 1 (along quantified differential equation $c' = 1$, which is equivalent to $\forall i : A\ c' = 1$ where i does not occur in c). Then $\alpha \oplus c' = 1$ is defined inductively as follows:*

1. $\forall i : A\ f(\boldsymbol{s}) := \theta \oplus c' = 1 := \forall i : A\ f(\boldsymbol{s}) := \theta$
2. $\forall i : A\ f(\boldsymbol{s})' = \theta\ \&\ \chi \oplus c' = 1 := \forall i : A\ f(\boldsymbol{s})' = \theta, c' = 1\ \&\ \chi$
3. $?\chi \oplus c' = 1 := ?\chi$
4. $\alpha \cup \beta \oplus c' = 1 := \alpha \oplus c' = 1 \cup \beta \oplus c' = 1$
5. $\alpha; \beta \oplus c' = 1 := \alpha \oplus c' = 1; \beta \oplus c' = 1$
6. $\alpha^* \oplus c' = 1 := (\alpha \oplus c' = 1)^*$.

Inherited Non-temporal Rules. The QdMTL calculus inherits the (non-temporal) $\mathsf{Qd}\mathcal{L}$ proof rules. For propositional logic, standard rules *ax–cut* are listed in Fig. 1. Rules $[;]$–$\langle?\rangle$ work similar to those in [13]. Rules $[']$, $\langle'\rangle$ handle continuous evolutions for quantified differential equations with first-order definable solutions. Rules $[:=]$–$\langle:*\rangle$ handle discrete changes for quantified assignments. Axiom *ex* expresses that, for sort $A \neq \mathbb{R}$, there always is a new object n that

$$(ax)\ \frac{}{\phi \to \phi} \qquad (\neg\mathrm{r})\ \frac{\phi \to}{\to \neg\phi} \qquad (\neg\mathrm{l})\ \frac{\to \phi}{\neg\phi \to} \qquad (\wedge\mathrm{r})\ \frac{\to \phi \quad \to \psi}{\to \phi \wedge \psi} \qquad (\wedge\mathrm{l})\ \frac{\phi, \psi \to}{\phi \wedge \psi \to} \qquad (cut)\ \frac{\to \phi \quad \phi \to}{\to}$$

$$([;])\ \frac{[\alpha][\beta]\phi}{[\alpha;\beta]\phi} \qquad (\langle;\rangle)\ \frac{\langle\alpha\rangle\langle\beta\rangle\phi}{\langle\alpha;\beta\rangle\phi} \qquad ([\cup])\ \frac{[\alpha]\phi \wedge [\beta]\phi}{[\alpha \cup \beta]\phi} \qquad (\langle\cup\rangle)\ \frac{\langle\alpha\rangle\phi \vee \langle\beta\rangle\phi}{\langle\alpha \cup \beta\rangle\phi} \qquad ([?])\ \frac{\chi \to \phi}{[?\chi]\phi} \qquad (\langle?\rangle)\ \frac{\chi \wedge \phi}{\langle?\chi\rangle\phi}$$

$$(['])\ \frac{\forall t \geq 0((\forall 0 \leq \tilde{t} \leq t[\forall i : A\ \mathcal{S}(\tilde{t})]\chi) \to [\forall i : A\ \mathcal{S}(t)]\phi)}{[\forall i : A\ f(\boldsymbol{s})' = \theta\ \&\ \chi]\phi}\ {}^{1} \qquad (\langle'\rangle)\ \frac{\exists t \geq 0((\forall 0 \leq \tilde{t} \leq t\langle\forall i : A\ \mathcal{S}(\tilde{t})\rangle\chi) \wedge \langle\forall i : A\ \mathcal{S}(t)\rangle\phi)}{\langle\forall i : A\ f(\boldsymbol{s})' = \theta\ \&\ \chi\rangle\phi}\ {}_{1}$$

$$([:=])\ \frac{\mathsf{if}\ \exists i : A\ \boldsymbol{s} = [\mathcal{A}]\boldsymbol{u}\ \mathsf{then}\ \forall i : A\ (\boldsymbol{s} = [\mathcal{A}]\boldsymbol{u} \to \phi(\theta))\ \mathsf{else}\ \phi(f([\mathcal{A}]\boldsymbol{u}))\ \mathsf{fi}}{\phi([\forall i : A\ f(\boldsymbol{s}) := \theta]f(\boldsymbol{u}))}\ {}^{2}$$

$$(\langle:=\rangle)\ \frac{\mathsf{if}\ \exists i : A\ \boldsymbol{s} = \langle\mathcal{A}\rangle\boldsymbol{u}\ \mathsf{then}\ \exists i : A\ (\boldsymbol{s} = \langle\mathcal{A}\rangle\boldsymbol{u} \wedge \phi(\theta))\ \mathsf{else}\ \phi(f(\langle\mathcal{A}\rangle\boldsymbol{u}))\ \mathsf{fi}}{\phi(\langle\forall i : A\ f(\boldsymbol{s}) := \theta\rangle f(\boldsymbol{u}))}\ {}_{2}$$

$$(skip)\ \frac{\Upsilon([\forall i : A\ f(\boldsymbol{s}) := \theta]\boldsymbol{u})}{[\forall i : A\ f(\boldsymbol{s}) := \theta]\Upsilon(\boldsymbol{u})}\ {}^{3} \qquad ([:*])\ \frac{\forall j : A\ \phi(\theta)}{[\forall j : A\ n := \theta]\phi(n)} \qquad (\langle:*\rangle)\ \frac{\exists j : A\ \phi(\theta)}{\langle\forall j : A\ n := \theta\rangle\phi(n)} \qquad (ex)\ \frac{true}{\exists n : A\ \mathsf{E}(n) = 0}$$

$$(\forall\mathrm{r})\ \frac{\Gamma \to \phi(f(X_1, \ldots, X_n)), \Delta}{\Gamma \to \forall x\phi(x), \Delta}\ {}^{4} \qquad (\exists\mathrm{r})\ \frac{\Gamma \to \phi(\theta), \exists x\phi(x), \Delta}{\Gamma \to \exists x\phi(x), \Delta}\ {}^{5} \qquad (\forall\mathrm{l})\ \frac{\Gamma, \phi(\theta), \forall x\phi(x) \to \Delta}{\Gamma, \forall x\phi(x) \to \Delta}\ {}^{5} \qquad (\exists\mathrm{l})\ \frac{\Gamma, \phi(f(X_1, \ldots, X_n)) \to \Delta}{\Gamma, \exists x\phi(x) \to \Delta}\ {}^{4}$$

$$(\mathrm{i}\forall)\ \frac{\mathrm{QE}(\forall X, Y(\mathsf{if}\ \boldsymbol{s} = \boldsymbol{t}\ \mathsf{then}\ \Phi(X) \to \Psi(X)\ \mathsf{else}\ \Phi(X) \to \Psi(Y)\ \mathsf{fi}))}{\Phi(f(\boldsymbol{s})) \to \Psi(f(\boldsymbol{t}))}\ {}^{6} \qquad (\mathrm{i}\exists)\ \frac{\mathrm{QE}(\exists X \bigwedge_i(\Phi_i \to \Psi_i))}{\Phi_1 \to \Psi_1\ \ \ldots\ \ \Phi_n \to \Psi_n}\ {}^{7}$$

$$([]gen)\ \frac{\phi \to \psi}{\Gamma, [\alpha]\phi \to [\alpha]\psi, \Delta} \qquad (\langle\rangle gen)\ \frac{\phi \to \psi}{\Gamma, \langle\alpha\rangle\phi \to \langle\alpha\rangle\psi, \Delta} \qquad (ind)\ \frac{\phi \to [\alpha]\phi}{\Gamma, \phi \to [\alpha^*]\phi, \Delta} \qquad (con)\ \frac{v > 0 \wedge \varphi(v) \to \langle\alpha\rangle\varphi(v-1)}{\Gamma, \exists v\, \varphi(v) \to \langle\alpha^*\rangle\exists v \leq 0\, \varphi(v), \Delta}\ {}^{8}$$

$$(DI)\ \frac{\chi \to [\forall i : A\ f(\boldsymbol{s})' := \theta]D(\phi)}{\phi \to [\forall i : A\ f(\boldsymbol{s})' = \theta\ \&\ \chi]\phi}\ {}^{9} \qquad (DC)\ \frac{\phi \to [\forall i : A\ f(\boldsymbol{s})' = \theta\ \&\ \chi]\psi \quad \phi \to [\forall i : A\ f(\boldsymbol{s})' = \theta\ \&\ \chi \wedge \psi]\phi}{\phi \to [\forall i : A\ f(\boldsymbol{s})' = \theta\ \&\ \chi]\phi}$$

$$([\cup]\Box)\ \frac{[\alpha]\Box\phi \wedge [\beta]\Box\phi}{[\alpha \cup \beta]\Box\phi}\ {}^{10} \qquad (\langle\cup\rangle\Diamond)\ \frac{\langle\alpha\rangle\Diamond\phi \vee \langle\beta\rangle\Diamond\phi}{\langle\alpha \cup \beta\rangle\Diamond\phi}\ {}^{11} \qquad ([;]\Box)\ \frac{[\alpha]\Box\phi \wedge [\alpha][\beta]\Box\phi}{[\alpha;\beta]\Box\phi}\ {}^{10}$$

$$(\langle;\rangle\Diamond)\ \frac{\langle\alpha\rangle\Diamond\phi \vee \langle\alpha\rangle\langle\beta\rangle\Diamond\phi}{\langle\alpha;\beta\rangle\Diamond\phi}\ {}^{11} \qquad ([?]\Box)\ \frac{\phi}{[?\chi]\Box\phi}\ {}^{10} \qquad (\langle?\rangle\Diamond)\ \frac{\phi}{\langle?\chi\rangle\Diamond\phi}\ {}^{11}$$

$$([']\Box)\ \frac{[\forall i : A\ f(\boldsymbol{s})' = \theta\ \&\ \chi]\phi}{[\forall i : A\ f(\boldsymbol{s})' = \theta\ \&\ \chi]\Box\phi}\ {}^{10} \qquad (\langle'\rangle\Diamond)\ \frac{\langle\forall i : A\ f(\boldsymbol{s})' = \theta\ \&\ \chi\rangle\phi}{\langle\forall i : A\ f(\boldsymbol{s})' = \theta\ \&\ \chi\rangle\Diamond\phi}\ {}^{11}$$

$$([:=]\Box)\ \frac{\phi \wedge [\forall i : A\ f(\boldsymbol{s}) := \theta]\phi}{[\forall i : A\ f(\boldsymbol{s}) := \theta]\Box\phi}\ {}^{10} \qquad (\langle:=\rangle\Diamond)\ \frac{\phi \vee \langle\forall i : A\ f(\boldsymbol{s}) := \theta\rangle\phi}{\langle\forall i : A\ f(\boldsymbol{s}) := \theta\rangle\Diamond\phi}\ {}^{11}$$

$$([*^n]\Box)\ \frac{[\alpha;\alpha^*]\Box\phi}{[\alpha^*]\Box\phi}\ {}^{10} \qquad (\langle^{*n}\rangle\Diamond)\ \frac{\langle\alpha;\alpha^*\rangle\Diamond\phi}{\langle\alpha^*\rangle\Diamond\phi}\ {}^{11} \qquad ([^*]\Box)\ \frac{[\alpha^*][\alpha]\Box\phi}{[\alpha^*]\Box\phi}\ {}^{10} \qquad (\langle^*\rangle\Diamond)\ \frac{\langle\alpha^*\rangle\langle\alpha\rangle\Diamond\phi}{\langle\alpha^*\rangle\Diamond\phi}\ {}^{11}$$

$$([\cup]\Box_{\mathcal{I}})\ \frac{[c := 0; \alpha \cup \beta \oplus c' = 1; c' = 1]\Box(c \in \mathcal{I} \to \phi)}{[\alpha \cup \beta]\Box_{\mathcal{I}}\,\phi}\ {}^{10\,12} \qquad (\langle\cup\rangle\Diamond_{\mathcal{I}})\ \frac{\langle c := 0; \alpha \cup \beta \oplus c' = 1; c' = 1\rangle\Diamond(c \in \mathcal{I} \wedge \phi)}{\langle\alpha \cup \beta\rangle\Diamond_{\mathcal{I}}\,\phi}\ {}^{11\,12}$$

$$([;]\Box_{\mathcal{I}})\ \frac{[c := 0; \alpha; \beta \oplus c' = 1; c' = 1]\Box(c \in \mathcal{I} \to \phi)}{[\alpha;\beta]\Box_{\mathcal{I}}\,\phi}\ {}^{10\,12} \qquad ([;]\Diamond_{\mathcal{I}})\ \frac{\langle c := 0; \alpha; \beta \oplus c' = 1; c' = 1\rangle\Diamond(c \in \mathcal{I} \wedge \phi)}{\langle\alpha;\beta\rangle\Diamond_{\mathcal{I}}\,\phi}\ {}^{11\,12}$$

$$([?]\Box_{\mathcal{I}})\ \frac{[c := 0; ?\chi \oplus c' = 1; c' = 1]\Box(c \in \mathcal{I} \to \phi)}{[?\chi]\Box_{\mathcal{I}}\,\phi}\ {}^{10\,12} \qquad (\langle?\rangle\Diamond_{\mathcal{I}})\ \frac{\langle c := 0; ?\chi \oplus c' = 1; c' = 1\rangle\Diamond(c \in \mathcal{I} \wedge \phi)}{\langle?\chi\rangle\Diamond_{\mathcal{I}}\,\phi}\ {}^{11\,12}$$

$$([']\Box_{\mathcal{I}})\ \frac{[c := 0; \forall i : A\ f(\boldsymbol{s})' = \theta\ \&\ \chi \oplus c' = 1; c' = 1]\Box(c \in \mathcal{I} \to \phi)}{[\forall i : A\ f(\boldsymbol{s})' = \theta\ \&\ \chi]\Box_{\mathcal{I}}\,\phi}\ {}^{10\,12}$$

$$(\langle'\rangle\Diamond_{\mathcal{I}})\ \frac{\langle c := 0; \forall i : A\ f(\boldsymbol{s})' = \theta\ \&\ \chi \oplus c' = 1; c' = 1\rangle\Diamond(c \in \mathcal{I} \wedge \phi)}{\langle\forall i : A\ f(\boldsymbol{s})' = \theta\ \&\ \chi\rangle\Diamond_{\mathcal{I}}\,\phi}\ {}^{11\,12}$$

$$([:=]\Box_{\mathcal{I}})\ \frac{[c := 0; \forall i : A\ f(\boldsymbol{s}) := \theta \oplus c' = 1; c' = 1]\Box(c \in \mathcal{I} \to \phi)}{[\forall i : A\ f(\boldsymbol{s}) := \theta]\Box_{\mathcal{I}}\,\phi}\ {}^{10\,12}$$

$$(\langle:=\rangle\Diamond_{\mathcal{I}})\ \frac{\langle c := 0; \forall i : A\ f(\boldsymbol{s}) := \theta \oplus c' = 1; c' = 1\rangle\Diamond(c \in \mathcal{I} \wedge \phi)}{\langle\forall i : A\ f(\boldsymbol{s}) := \theta\rangle\Diamond_{\mathcal{I}}\,\phi}\ {}^{11\,12}$$

$$([^*]\Box_{\mathcal{I}})\ \frac{[c := 0; \alpha^* \oplus c' = 1; c' = 1]\Box(c \in \mathcal{I} \to \phi)}{[\alpha^*]\Box_{\mathcal{I}}\,\phi}\ {}^{10\,12} \qquad (\langle^*\rangle\Diamond_{\mathcal{I}})\ \frac{\langle c := 0; \alpha^* \oplus c' = 1; c' = 1\rangle\Diamond(c \in \mathcal{I} \wedge \phi)}{\langle\alpha^*\rangle\Diamond_{\mathcal{I}}\,\phi}\ {}^{11\,12}$$

[1] $t, \tilde{t}$ are new variables, $\forall i : A\ \mathcal{S}(t)$ is the quantified assignment $\forall i : A\ f(\boldsymbol{s}) := y_{\boldsymbol{s}}(t)$ with solutions $y_{\boldsymbol{s}}(t)$ of the (injective) differential equations and $f(\boldsymbol{s})$ as initial values. See [22, 23] for the definition of a *injective* quantified assignment or quantified differential equation.

[2] Occurrence $f(\boldsymbol{u})$ in $\phi(f(\boldsymbol{u}))$ is not in scope of a modality (admissible substitution) and we abbreviate assignment $\forall i : A\ f(\boldsymbol{s}) := \theta$ by $\mathcal{A}$, which is assumed to be injective.

[3] $f \neq \Upsilon$ and the quantified assignment $\forall i : A\ f(\boldsymbol{s}) := \theta$ is injective. The same rule applies for $\langle\forall i : A\ f(\boldsymbol{s}) := \theta\rangle$ instead of $[\forall i : A\ f(\boldsymbol{s}) := \theta]$.

[4] f is a new (Skolem) function and $X_1, \ldots, X_n$ are all free logical variables of $\forall x\phi(x)$.

[5] θ is an abbreviate term, often a new logical variable.

[6] X, Y are new variables of sort $\mathbb{R}$. QE needs to be applicable in the premises.

[7] Among all open branches, the free (existential) logical variable X of sort $\mathbb{R}$ only occurs in the branch $\Phi_i \to \Psi_i$. QE needs to be defined for the formula in the premises, especially, no Skolem dependencies on X occur.

[8] logical variable v does not occur in α.

[9] The operator D, as defined in [24], is used to computer syntactic total derivations of formulas algebraically.

[10] $\Box$ is the abbreviation for the metric temporal modality $\Box_{[0,+\infty)}$.

[11] $\Diamond$ is the abbreviation for the metric temporal modality $\Diamond_{[0,+\infty)}$.

[12] c is a new variable of sort $\mathbb{R}$ (clock variable).

Figure 1: Rule schemata of the proof calculus for QdMTL

has not been created yet ($\mathsf{E}(n) = 0$), because domains are infinite. The quantifier rules ∀r–i∃ combine quantifier handling of many-sorted logic based on instantiation with theory reasoning by *quantifier elimination* (QE) for the theory of reals. The global rules $[]gen$, $\langle\rangle gen$ are Gödel generalization rules and *ind* is an induction schema for loops with *inductive invariant* ϕ [13]. Similarly, *con* generalizes Harel's convergence rule [13] to the hybrid case with decreasing variant φ [21]. *DI* and *DC* are rules for quantified differential equations with *quantified differential invariants* [24].

Metric Temporal Rules. The metric temporal rules $[\cup]_{\mathcal{I}}$–$\langle^*\rangle_{\mathcal{I}}$ in Fig. 1 for the QdMTL calculus convert metric temporal statements into equivalent non-metric temporal formulas. This conversion is based on the idea of referring metric temporal constraints to the reading on clocks, i.e. variables changing with constant slope 1 (along quantified differential equation $c' = 1$). Clocks are reset by the quantified assignment $c := 0$, which is equivalent to $\forall i : A\ c := 0$ where i does not occur in c, before quantified hybrid programs start executing. They are used to keep track of the progress of time advancing with constant slope 1.

Non-metric Temporal Rules. The non-metric temporal rules $[\cup]$–$\langle^*\rangle$ in Fig. 1 for the QdMTL calculus successively transform non-metric temporal specifications of quantified hybrid programs into non-temporal QdℒL formulas. The idea underlying this transformation is to decompose quantified hybrid programs and recursively augment intermediate state transitions with appropriate specifications.

Rule $[\cup]$ decomposes invariants of $\alpha \cup \beta$ (i.e., $[\alpha \cup \beta]\Box\phi$ holds) into an invariant of α (i.e., $[\alpha]\Box\phi$) and an invariant of β (i.e., $[\beta]\Box\phi$)). The QdℒL rule $[\cup]$ can actually be generalized to apply to formulas of the form $[\alpha \cup \beta]\pi$ where π is an arbitrary trace formula, and not just a state formula as in QdℒL. Rule $[;]$ decomposes invariants of $\alpha;\beta$ (i.e., $[\alpha;\beta]\Box\phi$ holds) into an invariant of α (i.e., $[\alpha]\Box\phi$) and an invariant of β that holds when β is started in *any* final state of α (i.e., $[\alpha]([\beta]\Box\phi)$). Its difference with the QdℒL rule $[;]$ thus is that the QdMTL rule $[;]$ also checks safety invariant ϕ at the symbolic states in between the execution of α and β, and recursively so because of the temporal modality $\Box$. Rule $[:=]$ expresses that invariants of quantified assignments need to hold before and after the discrete change (similarly for $[?]$, except that tests do not lead to a state change, so ϕ holding before the test is all there is to it). Rule $[']$ can directly reduce invariants of continuous evolutions to non-temporal formulas as restrictions of solutions of quantified differential equations are themselves solutions of different duration and thus already included in the continuous evolutions of $\forall i : A\ f(\boldsymbol{s})' = \theta$. The (optional) iteration rule $[^{*n}]$ can partially unwind loops. It relies on rule $[;]$. The dual rules $\langle\cup\rangle\Diamond$, $\langle;\rangle\Diamond$, $\langle:=\rangle\Diamond$, $\langle?\rangle\Diamond$, $\langle'\rangle\Diamond$, $\langle^{*n}\rangle\Diamond$ work similarly.

Rules $[^*]$ and $\langle^*\rangle\Diamond$ actually define temporal properties of loops inductively. Rule $[^*]$ expresses that ϕ holds at all times during repetitions of α (i.e., $[\alpha^*]\Box\phi$) iff, after repeating α any number of times, ϕ holds at all times during one execution of α (i.e., $[\alpha^*]([\alpha]\Box\phi)$). Dually, $\langle^*\rangle\Diamond$ expresses that α holds at some time during repetitions of α (i.e., $\langle\alpha^*\rangle\Diamond\phi$) iff, after some number of repetitions of α, formula ϕ holds at some point during one execution of α (i.e., $\langle\alpha^*\rangle(\langle\alpha\rangle\Diamond\phi)$). In this context, the non-temporal modality $\langle\alpha^*\rangle$ can be thought of as skipping over to the iteration of α during which ϕ actually occurs, as expressed by the nested QdMTL formula $\langle\alpha\rangle\Diamond\phi$. The inductive definition rules $[^*]$ and $\langle^*\rangle\Diamond$ completely reduce temporal properties of loops to QdMTL properties of standard non-temporal QdℒL modalities such that standard induction (*ind*) or convergence (*con*) rules, as listed in Fig. 1, can be used for the outer non-temporal modality of the loop. Hence, after applying the inductive loop definition rules $[^*]$ and $\langle^*\rangle\Diamond$, the standard QdℒL loop invariant and variant rules can be used for verifying temporal properties of

loops without change, except that the postcondition contains temporal modalities.

5.2 Soundness and Completeness

Soundness. The following result shows that verification with the QdMTL calculus always produces correct results about the metric temporal behavior of distributed hybrid systems, i.e., the QdMTL calculus is sound.

Theorem 7 (Soundness of QdMTL)**.** *The* QdMTL *calculus is sound, i.e., every* QdMTL *(state) formula that can be proven is valid.*

Incompleteness of QdMTL. In [22, 23] it has been shown that the verification problem for distributed hybrid systems has *three independent sources* of undecidability. Both the discrete and continuous fragments of Qd$\mathcal{L}$ are subject to Gödel's incompleteness theorem. The fragment with only structural and dimension-changing dynamics is not effective either, because it can encode two-counter machines. Hence, Qd$\mathcal{L}$ cannot be effectively axiomatizable. Since QdMTL is a conservative extension of Qd$\mathcal{L}$, those results lift to QdMTL. Therefore, the discrete, continuous, and structural fragments of QdMTL, even if only containing non-temporal formulas are non-axiomatizable. In particular, QdMTL is non-axiomatizable.

Relative Completeness. The Qd$\mathcal{L}$ calculus has been proved to be complete relative to the *first-order logic of quantified differential equations* (FOQD), i.e., first-order real arithmetic augmented with formulas expressing properties of quantified differential equations, that is, Qd$\mathcal{L}$ formulas of the form $[\forall i : A\ f(\boldsymbol{s})' = \theta \,\&\, \chi]F$ with a first-order formula F [23]. Due to the modular construction of the QdMTL calculus, we can lift the relative completeness result from Qd$\mathcal{L}$ to QdMTL. We essentially show that QdMTL is complete relative to Qd$\mathcal{L}$, which directly implies that QdMTL is even complete relative to FOQD. Again, we restrict our attention to homogeneous combinations of path and trace quantifiers like $[\alpha]\Box_{\mathcal{I}}\,\phi$ or $\langle\alpha\rangle\Diamond_{\mathcal{I}}\,\phi$.

Theorem 8 (Relative Completeness of QdMTL)**.** *The* QdMTL *calculus in Fig. 1 is complete relative to* FOQD*, i.e., every valid* QdMTL *formulas can be derived from* FOQD *tautologies.*

This result shows that metric temporal, non-metric temporal, and non-temporal properties of distributed hybrid systems can be proven to exactly the same extent to which properties of quantified differential equations can be proven. It also gives a formal justification that the QdMTL calculus reduces metric temporal properties to non-temporal Qd$\mathcal{L}$ properties.

6 Conclusions and Future Work

For reasoning about distributed hybrid systems, we have introduced a metric temporal dynamic logic, QdMTL, with modal path quantifiers over traces and metric temporal quantifiers along the traces. It combines the capabilities of quantified differential dynamic logic to reason about possible distributed hybrid system behavior with the power of metric temporal logic in reasoning about the behavior along time intervals of traces. We have presented a proof calculus for verifying metric temporal safety specifications of quantified hybrid programs in QdMTL, which, to the best of our knowledge, is the first verification approach that can handle metric temporal statements about distributed hybrid systems. Our sequent calculus for QdMTL is a completely modular combination of metric temporal, non-metric temporal, and non-temporal reasoning. Furthermore, We have proven our calculus to be a sound and complete axiomatization relative to quantified differential equations.

We are currently extending a verification tool for distributed hybrid systems, which is an automated theorem prover called KeYmaeraD [26], to cover the full QdMTL calculus. One direction for future work is to extend QdMTL with parametric temporal operators [2] to formulate the *quantified differential parametric metric temporal dynamic logic* (QdPMTL) for specifying and verifying parametric temporal properties of distributed hybrid systems. We would also like to symbolically synthesis parametric safety timing constraints using the QdPMTL calculus. Another direction is to move toward the deductive verification of temporal properties of distributed stochastic hybrid systems.

References

[1] R. Alur, C. Courcoubetis, and D. L. Dill. Model-checking for real-time systems. In *LICS*, pages 414–425. IEEE Computer Society, 1990.

[2] R. Alur, K. Etessami, S. La Torre, and D. Peled. Parametric temporal logic for "model measuring". In J. Wiedermann, P. van Emde Boas, and M. Nielsen, editors, *ICALP*, volume 1644 of *LNCS*, pages 159–168. Springer, 1999.

[3] B. Beckert and S. Schlager. A sequent calculus for first-order dynamic logic with trace modalities. In R. Goré, A. Leitsch, and T. Nipkow, editors, *IJCAR*, volume 4130 of *LNCS*, pages 626–641. Springer, 2001.

[4] J. M. Davoren, V. Coulthard, N. Markey, and T. Moor. Non-deterministic temporal logics for general flow systems. In R. Alur and G. J. Pappas, editors, *HSCC*, volume 2993 of *LNCS*, pages 280–295. Springer, 2004.

[5] J. M. Davoren and A. Nerode. Logics for hybrid systems. *Proceedings of the IEEE*, 88(7):985–1010, July 2000.

[6] A. Deshpande, A. Göllü, and P. Varaiya. SHIFT: A formalism and a programming language for dynamic networks of hybrid automata. In *Hybrid Systems*, pages 113–133, 1996.

[7] G. Dowek, C. Muñoz, and V. A. Carreño. Provably safe coordinated strategy for distributed conflict resolution. In *AIAA Proceedings, AIAA-2005-6047*, pages 278–292, 2005.

[8] E. A. Emerson and E. M. Clarke. Using branching time temporal logic to synthesize synchronization skeletons. *Sci. Comput. Program*, 2(3):241–266, 1982.

[9] E. A. Emerson and J. Y. Halpern. "Sometimes" and "Not Never" revisited: on branching versus linear time temporal logic. *J. ACM*, 33(1):151–178, 1986.

[10] G. Fainekos and A. Girardand G. J. Pappas. Temporal logic verification using simulation. In E. Asarin and P. Bouyer, editors, *FORMATS*, volume 4202 of *LNCS*, pages 171–186. Springer, 2006.

[11] G. Fainekos and G. Pappas. Robustness of temporal logic specications for continuous-time signals. *Theoretical Computer Science*, 410(42):4262–4291, 2009.

[12] T. Flaminio and E. B. P. Tiezzi. On metric temporal lukasiewicz logic. *Electr. Notes Theor. Comput. Sci.*, 246:71–85, 2009.

[13] D. Harel, D. Kozen, and J. Tiuryn. *Dynamic logic*. MIT Press, Cambridge, 2000.

[14] T. A. Henzinger, X. Nicollin, J. Sifakis, and S. Yovine. Symbolic model checking for real-time systems. In *LICS*, pages 394–406. IEEE Computer Society, 1992.

[15] A. Hsu, F. Eskafi, S. Sachs, and P. Varaiya. Design of platoon maneuver protocols for IVHS. Technical Report PATH Research Report UCB-ITS-PRR-91-6, UC Berkeley, 1991.

[16] R. Koymans. Specifying real-time properties with metric temporal logic. *Real-Time Systems*, 2(4):255–299, 1990.

[17] F. Kratz, O. Sokolsky, G. J. Pappas, and I. Lee. R-Charon, a modeling language for reconfigurable hybrid systems. In *HSCC*, pages 392–406, 2006.

[18] O. Maler and D. Nickovic. Monitoring temporal properties of continuous signals. In Y. Lakhnech and S. Yovine, editors, *FORMATS-FTRTFT*, volume 3253 of *LNCS*, pages 152–166. Springer, 2004.

[19] F. Montagna, G. Michele Pinna, and E. B. P. Tiezzi. A cut-free proof system for bounded metric temporal logic over a dense time domain. *Math. Log. Q.*, 46(2):171–182, 2000.

[20] E. Plaku, L. E. Kavraki, and M. Y. Vardi. Falsication of LTL safety properties in hybrid systems. In S. Kowalewski and A. Philippou, editors, *TACAS*, volume 5505 of *LNCS*, pages 368–382. Springer, 2009.

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

[22] A. Platzer. Quantified differential dynamic logic for distributed hybrid systems. In A. Dawar and H. Veith, editors, *CSL*, volume 6247 of *LNCS*, pages 469–483. Springer, 2010.

[23] A. Platzer. Quantified differential dynamic logic for distributed hybrid systems. Technical Report CMU-CS-10-126, School of Computer Science, Carnegie Mellon University, Pittsburgh, PA, May 2010.

[24] A. Platzer. Quantified differential invariants. In E. Frazzoli and R. Grosu, editors, *HSCC*, pages 63–72. ACM, 2011.

[25] V. R. Pratt. Process logic. In *POPL*, pages 93–100, 1979.

[26] D. W. Renshaw, S. M. Loos, and A. Platzer. Distributed theorem proving for distributed hybrid systems. In S. Qin and Z. Qiu, editors, *ICFEM*, volume 6991 of *LNCS*, pages 356–371. Springer, 2011.

[27] S. Sankaranarayanan and G. Fainekos. Falsification of temporal properties of hybrid systems using the cross-entropy method. In T. Dang and I. M. Mitchell, editors, *HSCC*, pages 125–134. ACM, 2012.